



# **ESCUELA SUPERIOR DE INGENIERÍA**

## **INGENIERO TÉCNICO INDUSTRIAL EN ELECTRICIDAD Y ELECTRÓNICA INDUSTRIAL**

### **ROBOT AUTÓNOMO DE VIGILANCIA CON PROCESAMIENTO DE IMÁGENES EN TIEMPO REAL**

**DEPARTAMENTO:** INGENIERÍA EN AUTOMÁTICA, ELECTRÓNICA,  
ARQUITECTURA Y REDES DE COMPUTADORES

**DIRECTORES DEL PROYECTO:** VÍCTOR MANUEL SÁNCHEZ CORBACHO  
ARTURO MORGADO ESTÉVEZ

**AUTOR DEL PROYECTO:** JOSE MANUEL REINA RODRÍGUEZ

Cádiz, Septiembre de 2012

Fdo. Jose Manuel Reina Rodríguez

*No hay una cosa más difícil que otra, solo se diferencian  
en el número de horas de trabajo que conllevan.*

# Índice general

---

1. Memoria .....	1
1.1. Objeto .....	2
1.2. Alcance .....	3
1.3. Antecedentes .....	3
1.3.1. ¿Por qué este proyecto? .....	3
1.3.2. Robótica y Visión Artificial .....	4
1.3.2.1. Principios de robótica .....	4
1.3.2.1.1. Historia de la robótica .....	4
1.3.2.1.2. Tipos de robots .....	5
1.3.2.1.3. ¿Qué es un robot? .....	5
1.3.2.1.4. Robótica en la actualidad .....	5
1.3.2.2. Visión Artificial .....	6
1.3.2.2.1. Alcance de la visión artificial .....	7
1.3.3. Estado del arte .....	8
1.4. Normas y referencias .....	9
1.4.1. Normas .....	9
1.4.2. Bibliografía .....	9
1.4.3. Recursos empleados .....	10
1.4.3.1. Información de internet .....	10
1.4.3.2. Software utilizado .....	11
1.5. Definiciones y abreviaturas .....	14
1.6. Requisitos de diseño .....	16
1.7. Análisis de soluciones .....	17
1.7.1. Hardware .....	17
1.7.1.1. Placas de desarrollo usadas .....	17

1.7.1.2. Movimiento.....	21
1.7.1.3. Motor y controlador .....	22
1.7.1.4. Estructura.....	23
1.7.1.5. Cabeza articulada.....	24
1.7.1.6. Cámara .....	24
1.7.1.7. Articulación.....	26
1.7.1.8. Sensores .....	26
1.7.1.9. Conexión Wi-Fi.....	27
1.7.1.10. Alimentación.....	28
1.7.1.10.1. Tipos de baterías.....	28
1.7.2. Software.....	30
1.7.2.1. Software microcontrolador.....	30
1.7.2.2. Software DSP .....	30
1.7.2.3. Software Ordenador .....	30
1.8. Solución adoptada.....	31
1.8.1. Decisiones tomadas .....	31
1.8.1.1. Estructura.....	31
1.8.1.2. Movimiento.....	31
1.8.1.3. Motor .....	31
1.8.1.4. Electrónica usada .....	32
1.8.1.5. Cabeza articulada.....	32
1.8.1.6. Sensores .....	33
1.8.1.7. Conexión Wi-Fi.....	33
1.8.1.8. Alimentación.....	34
1.8.2. Montaje.....	35
1.8.2.1. Base .....	35
1.8.2.2. Plataformas superiores .....	40
1.8.2.3. Montaje final.....	51
1.8.2.4. Cabeza articulada.....	54
1.8.2.5. Sensores infrarrojos .....	56
1.8.2.6. Ventilador.....	57
1.8.2.7. Módulo Wi-Fi.....	57
1.8.2.8. Batería y fuente de alimentación.....	58
1.8.3. Descripción del sistema .....	61
1.8.3.1. Definición general del sistema .....	61
1.8.3.2. Grupo 1 .....	65



1.8.3.2.1. Descripción general .....	65
1.8.3.2.1.1. Composición .....	65
1.8.3.2.1.2. Esquema .....	65
1.8.3.2.2. Funcionamiento global.....	66
1.8.3.2.3. Funcionamiento individual.....	66
1.8.3.2.3.1. Ordenador .....	66
1.8.3.2.3.2. Router.....	67
1.8.3.2.3.3. Joystick.....	67
1.8.3.2.3.4. Software .....	70
1.8.3.3. Grupo 2.....	71
1.8.3.3.1. Descripción general .....	71
1.8.3.3.1.1. Composición .....	71
1.8.3.3.1.2. Esquema .....	71
1.8.3.3.2. Funcionamiento global.....	72
1.8.3.3.3. Funcionamiento de cada elemento.....	73
1.8.3.3.3.1. Webcam .....	73
1.8.3.3.3.2. Adaptador Wi-Fi USB.....	73
1.8.3.3.3.3. BeagleBoard-xM.....	74
1.8.3.3.3.4. Software .....	75
1.8.3.3.3.4.1. Funcionamiento general.....	75
1.8.3.3.3.4.2. Programación principal .....	75
1.8.3.3.3.4.3. Procesamiento de imágenes .....	77
1.8.3.3.3.4.4. Autonomía.....	78
1.8.3.3.3.4.5. Comunicación con grupo uno.....	78
1.8.3.3.3.4.6. Comunicación con grupo tres .....	78
1.8.3.4. Grupo 3.....	79
1.8.3.4.1. Descripción general .....	79
1.8.3.4.1.1. Composición .....	79
1.8.3.4.1.2. Esquema .....	79
1.8.3.4.2. Funcionamiento global.....	80
1.8.3.4.2.1. Órdenes.....	80
1.8.3.4.2.1.1. Orden de ejecución .....	80
1.8.3.4.2.1.2. Orden de devolución .....	80
1.8.3.4.3. Funcionamiento de cada elemento o elementos .....	81
1.8.3.4.3.1. El microcontrolador .....	81
1.8.3.4.3.2. Sensores .....	83

1.8.3.4.3.2.1. Sensores Infrarrojos .....	83
1.8.3.4.3.2.1.1. Conexión.....	83
1.8.3.4.3.2.1.2. Funcionamiento .....	84
1.8.3.4.3.2.1.3. Salida.....	85
1.8.3.4.3.2.2. Radar ultrasónico.....	85
1.8.3.4.3.2.2.1. Conexión.....	86
1.8.3.4.3.2.2.2. Funcionamiento.....	87
1.8.3.4.3.2.2.3. Salida.....	89
1.8.3.4.3.3. Módulo L298 y el motor .....	90
1.8.3.4.3.3.1. Módulo L298.....	90
1.8.3.4.3.3.1.1. Funcionamiento.....	91
1.8.3.4.3.3.1.2. Conexión.....	93
1.8.3.4.3.3.2. Motor.....	94
1.8.3.4.3.4. Servomotores .....	95
1.8.3.4.3.4.1. Funcionamiento.....	95
1.8.3.4.3.4.2. Conexión.....	96
1.8.3.4.3.5. Láseres.....	96
1.8.3.4.3.5.1. Funcionamiento.....	97
1.8.3.4.3.5.2. Conexión.....	97
1.8.3.4.3.6. Placa de enlace .....	97
1.8.3.4.3.7. Software global .....	100
1.8.3.4.3.8. Software individual.....	102
1.8.3.4.3.8.1. Motor y Servomotores.....	102
1.8.3.4.3.8.1.1. Servomotores .....	102
1.8.3.4.3.8.1.2. Motor.....	105
1.8.3.4.3.8.2. Radar.....	106
1.8.3.4.3.8.3. Sensores infrarrojos.....	107
1.8.3.4.3.8.4. Láseres.....	108
1.8.3.4.3.8.5. I2C .....	108
1.8.4. Alimentación.....	110
1.8.4.1. Batería .....	110
1.8.4.2. Fuente de alimentación .....	111
1.8.4.3. Reguladores de tensión .....	113
1.9. Líneas futuras de desarrollo .....	114
1.10. Planificación.....	115
1.11. Conclusiones.....	116

2. Anexos.....	118
2.1. Código .....	140
3. Planos .....	142
4. Presupuesto .....	159
4.1. Cuadro de precios.....	160
4.1.1. Módulo L298.....	160
4.1.2. Placa de ASA .....	160
4.1.3. Beagleboard-xM.....	160
4.1.4. Placa de enlace.....	160
4.1.5. Alimentación.....	161
4.1.5.1. Placa de alimentación .....	161
4.1.5.2. Cables de alimentación.....	161
4.1.5.3. Conjunto batería y cargador .....	162
4.1.6. Servomotores.....	162
4.1.7. Sensores .....	162
4.1.8. Cámara .....	163
4.1.9. Comunicación Wi-Fi .....	163
4.2. Presupuesto.....	163

# Índice de Figuras

Figura 1: Esquema global del objeto del proyecto .....	2
Figura 2: Imagen de la película "Metrópolis", 1926 .....	4
Figura 3: Androide ficticio .....	5
Figura 4: Ojo digital.....	6
Figura 5: Diversos Robots de vigilancia.....	8
Figura 6: Esquema general de la idea .....	16
Figura 7: Placa de desarrollo basada en LPC2148.....	18
Figura 8: Placa DSP TMS350C6713.....	19
Figura 9: Raspberry Pi.....	20
Figura 10: BeagleBoard-xM .....	20
Figura 11: Tipos de configuraciones según movimiento .....	21
Figura 12: Integrado L298.....	22
Figura 13: Módulo L298.....	22
Figura 14: Coche teledirigido original .....	23
Figura 15: Ejemplo de cabeza articulada .....	24
Figura 16: Primera cámara usada .....	25
Figura 17: Webcam Logitech c210 .....	25
Figura 18: Módulo Wifly-RN131-GSX.....	27
Figura 19: Articulación con Servomotores Hitec HS311.....	32
Figura 20: Adaptador Wi-Fi USB.....	33
Figura 21: Vehículo en su forma original .....	35
Figura 22: primer montaje para ASA (a).....	36
Figura 23: Primer montaje para ASA (b) .....	37
Figura 24: modificación realizada al servomotor.....	38
Figura 25: base chasis (a) .....	38
Figura 26: base chasis (b).....	39
Figura 27: base chasis (c) .....	39
Figura 28: primera idea de montaje .....	42
Figura 29: Primer prototipo plataforma principal, cartón. (a) .....	43
Figura 30: Primer prototipo plataforma principal, cartón (b) .....	44
Figura 31: Primer prototipo plataforma principal, cartón.....	44
Figura 32: Plataformas con pisos 1º y 2º. (a).....	45
Figura 33: Plataformas con pisos 1º y 2º. (b) .....	45
Figura 34: Plataformas con pisos 1º y 2º (c).....	46
Figura 35: Colocación de sensores infrarrojos.....	47
Figura 36: Diseño plataforma principal .....	48
Figura 37: Base principal de metacrilato (a) .....	49
Figura 38: Base principal metacrilato (b) .....	49
Figura 39: Base principal de metacrilato .....	50
Figura 40: vista lateral derecho .....	51
Figura 41: vista lateral derecho .....	51
Figura 42: vista frontal.....	52
Figura 43: vista trasera .....	52
Figura 44: visión general.....	53
Figura 45: cabeza articulada frontal.....	54

Figura 46: cabeza articulada lateral.....	55
Figura 47: cabeza articulada superior .....	55
Figura 48: Disposición de los sensores infrarrojos .....	56
Figura 49: posición del ventilador.....	57
Figura 50: Colocación del módulo Wi-Fi .....	57
Figura 51: colocación de la batería.....	58
Figura 52: vista frontal de la fuente de alimentación .....	58
Figura 53: conector para la batería.....	59
Figura 54: cables de alimentación .....	59
Figura 55: conector para balanceador de la batería .....	60
Figura 56: logo Wi-Fi.....	62
Figura 57: Logo comunicación I2C.....	62
Figura 58: esquema general del sistema .....	63
Figura 59: aspecto físico del sistema.....	64
Figura 60: Esquema grupo 1.....	65
Figura 61: Ordenador portátil.....	66
Figura 62: Router utilizado.....	67
Figura 63: Joystick .....	68
Figura 64: adaptador Super Joy Box 7 .....	69
Figura 65: Logo QT Creator.....	70
Figura 66: Esquema grupo 2.....	71
Figura 67: cámara Logitech c210 .....	73
Figura 68: Adaptador Wi-Fi USB.....	73
Figura 69: BeagleBoard-xM .....	74
Figura 70: Logo librerías gstreamer .....	75
Figura 71: Pipeline seguido .....	76
Figura 72: Logo de OpenCV .....	77
Figura 73: Esquema grupo 3.....	79
Figura 74: placa con microcontrolador LPC2148 .....	81
Figura 75: depurador JTAG ARM USB Tiny .....	82
Figura 76: sensor infrarrojo.....	83
Figura 77: pines sensor infrarrojo .....	83
Figura 78: conexión del sensor infrarrojo.....	84
Figura 79: Gráfica de salida del sensor infrarrojo .....	85
Figura 80: radar ultrasónico .....	86
Figura 81: pines del radar .....	86
Figura 82: conexión del radar .....	87
Figura 83: esquema funcionamiento radar .....	88
Figura 84: tiempos de pulsos y respuestas en el radar .....	88
Figura 85: tiempos del radar .....	89
Figura 86: integrado L298 .....	90
Figura 87: módulo L298 .....	90
Figura 88: esquema interno L298.....	91
Figura 89: esquema puente en H.....	92
Figura 90: conexión módulo L298.....	93
Figura 91: Motor RS360-H .....	94
Figura 92: especificaciones del motor .....	94
Figura 93: servomotor Hitec HS311 y articulación.....	95
Figura 94: conexión servomotor .....	96

Figura 95: láser .....	96
Figura 96: conexión de los láseres.....	97
Figura 97: placa de enlace.....	98
Figura 98: circuito de conexión para el integrado PCA9306 .....	99
Figura 99: tabla de valores de resistencias pull-up .....	99
Figura 100: diagrama de flujo del microcontrolador.....	101
Figura 101: funcionamiento del pwm para los motores.....	102
Figura 102: diagrama de flujo de las interrupciones de los servomotores.....	103
Figura 103: diagrama de flujo de la llamada a modificar los servomotores .....	104
Figura 104: diagrama de flujo del motor.....	105
Figura 105: diagrama de flujo para leer el radar.....	106
Figura 106: diagrama de flujo para leer un sensor infrarrojo.....	107
Figura 107: diagrama de flujo para trabajar con un láser .....	108
Figura 108: diagrama de flujo del funcionamiento del i2c.....	109
Figura 109: batería LiPo 3S.....	110
Figura 110: tabla de intensidades necesarias .....	111
Figura 111: Funcionamiento fuente de alimentación .....	112
Figura 112: regulador de tensión conmutado .....	113

# Agradecimientos

*Quiero mostrar mi más sincero agradecimiento a las siguientes personas:*

*A mis tutores Arturo y Víctor, por estar cuando se les necesitaba y ayudarme siempre con tantos consejos sin ellos este proyecto no se habría podido llevar a cabo.*

*A mi novia Lucía, por todo el cariño, afecto y apoyo incondicional que siempre me proporciona por estar siempre a mi lado dándome ánimos para seguir adelante y por todos esos días en los que me quedaba trabajando en casa.*

*A mi cuñado Manolo, quien me ha ayudado en la resolución de muchísimos problemas estando siempre dispuesto a ello.*

*A mi hermana Reyes, por haber reñido al robot tantas veces diciéndole que me hiciese caso, creo que es por eso por lo que funciona bien.*

*A mis sobrinos Julio y Lucía, quienes no han parado de decirme lo mucho que les gustaba el coche.*

*A Manolo, Néstor y Antonio, por la ayuda y todos esos buenos momentos que me habéis hecho pasar. Cuando el trabajo se hace entre amigos se hace divertido.*

*A Manuel, gran programador, por esas explicaciones sobre procesamiento de imágenes.*

*A Jesús y Francisco Javier, dos amigos para siempre, con quienes no puedo parar de reírme.*

*A Roberto, por tanta ayuda en la fabricación de piezas en el laboratorio.*

*A Javier, por horas enteras de clase hablando sobre el robot.*

*Finalmente a todas aquellas personas con las que alguna vez hablé sobre mi proyecto y me aconsejaron sobre él.*





# 1. Memoria

## 1.1. Objeto

En el siguiente proyecto se muestra el diseño y montaje de un robot autónomo de vigilancia con capacidad para iniciar un movimiento evitando la colisión, capaz de detectar algún elemento determinado y pudiendo ser manejado desde un dispositivo portátil a través de una red Wi-Fi.

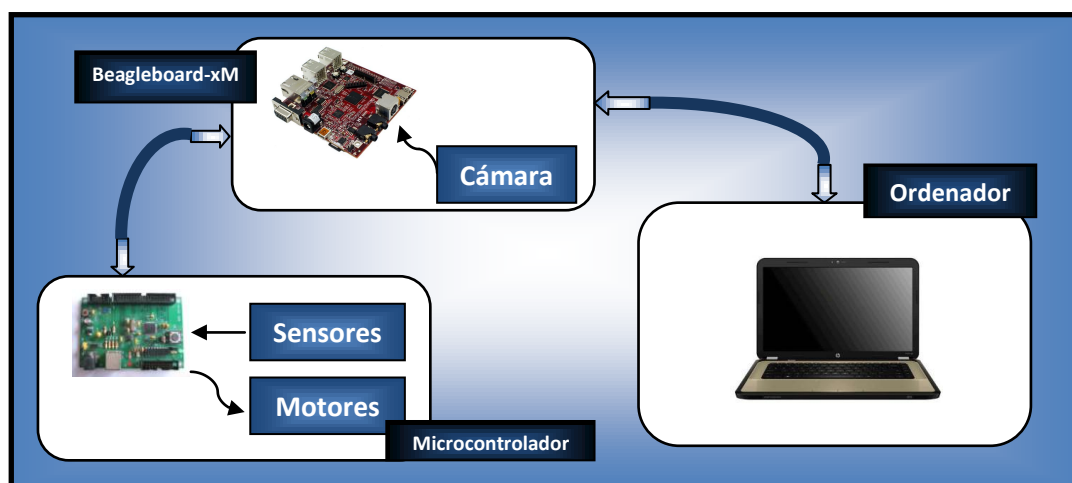
La implementación del sistema se ha llevado a cabo usando una placa de desarrollo basada en el microcontrolador NXP LPC2148 y un sistema embebido llamado “Beagleboard xM”. Estas dos placas operan conjuntamente para realizar las operaciones necesarias en cada elemento del sistema y poder realizar el objetivo que persiguen, cada una tiene sus funciones determinadas, pero siempre dependiendo la una de la otra.

El diseño incluye tanto el diseño hardware como el diseño software del mismo. Se ha llevado a cabo el montaje completo del robot partiendo de cero. Se han diseñado y fabricado las piezas que se han ido necesitando durante el proceso, o en su caso, adquiriéndolas ya fabricadas. El diseño del software ha consistido de tres grandes bloques, microcontrolador, sistema embebido y ordenador. Cada uno de estos bloques realiza una función diferente, pero todas interconectadas entre sí y funcionando a la vez.

Se dotará al sistema de sensores y partes móviles para poder llevar a cabo un movimiento pudiendo recibir información del entorno.

La diferencia de este sistema respecto a los ya existentes, según información disponible en la web, es su autonomía. Los prototipos existentes conocidos transmiten la información captada mediante cámara y sensores a un dispositivo de gran potencia, normalmente un ordenador, siendo este el que interpreta la información entrante y mandando órdenes de actuación al robot. El proyecto aquí presente no realiza tal tarea, es el mismo, el que capta información, la procesa, la evalúa, toma una decisión y la lleva a cabo.

La conexión a un dispositivo electrónico es indispensable, aunque se nombre como autónomo, debe estar conectado a algún punto a donde transmitir la señal con la información que va captando. Si no estuviese conectado a ningún sitio, no serviría de nada, ya que nadie tendría conocimiento de que está funcionando. Además de poder ser gobernado desde dicho dispositivo.



**Figura 1:** Esquema global del objeto del proyecto

## 1.2. Alcance

El presente proyecto pretende realizar un prototipo de robot capaz de llevar a cabo una vigilancia determinada, que sirva de apoyo a los sistemas de seguridad instalados en recintos cerrados, comunicado con el personal a través de una red Wi-Fi, pudiendo ser controlado y conociendo en todo momento el estado del mismo.

## 1.3. Antecedentes

A partir de las asignaturas impartidas “Informática Industrial” y “Aplicaciones de Sistemas y Automática” en la titulación de Ingeniero Técnico Industrial en Electrónica Industrial, en la que se lleva a cabo el aprendizaje de los microcontroladores, desde su implementación hasta su funcionamiento, y el interés personal de siempre por la electrónica, se quiso llevar a cabo la creación de un prototipo de robot móvil, que pudiese desempeñar alguna tarea por sí mismo.

Siguiendo consejos de mis tutores Arturo Morgado Estévez y Víctor Manuel Sánchez Corbacho, sin cuyos conocimientos de electrónica y programación esto no se podría desarrollar, a la vez que queriendo llevar a cabo una superación personal, el proyecto partió del control del motor y finaliza en el presente prototipo.

### 1.3.1. ¿Por qué este proyecto?

Muchas han sido las veces que me han preguntado el porqué he querido hacer de proyecto fin de carrera la creación de un prototipo de estas características, siempre he respondido de la misma manera cada vez: ¿Por qué no?.

El constante aprendizaje sobre microcontroladores, programación y electrónica en general, junto con una afán constante de superación personal, han terminado motivando la creación de un prototipo, moderno, tecnológico y divertido.

Descubriendo lo que se denomina “Inteligencia Artificial”, campo de mayor investigación en la robótica hoy en día, se quiso diseñar un robot, no que sea inteligente, puesto que es una palabra muy grande para mi proyecto, pero sí que disponga de una lógica de funcionamiento en la que sea capaz de tomar decisiones en base a unas condiciones.

A su vez, la Escuela Superior de Ingeniería, con los profesores que allí trabajan, los laboratorios y máquinas de que dispone, hacen del proyecto fin de carrera, una oportunidad de sacar a la luz todo lo aprendido y lo más importante, aprender más aún durante su ejecución.

## 1.3.2. Robótica y Visión Artificial

A continuación se dará una pequeña introducción a la robótica en general.

### 1.3.2.1. Principios de robótica

Creencias populares llevan tiempo pensando que el término robot se refiere a un aparato de aspecto humano que puede ejecutar ciertas tareas de forma similar a cómo las haría una persona. No más lejos de la realidad un robot es mucho más que eso.

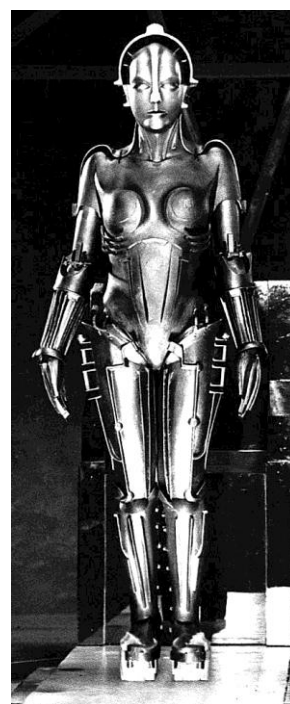
#### 1.3.2.1.1. Historia de la robótica

Por siglos, el ser humano ha construido máquinas que imitan partes del cuerpo humano. Los antiguos egipcios unieron brazos mecánicos a las estatuas de sus dioses, los griegos construyeron estatuas que operaban con sistemas hidráulicos, los cuales eran utilizados para fascinar a los adoradores de los templos.

La palabra robot, no es un término acuñado recientemente, el origen etimológico de esta palabra proviene del término checo "Robota" es decir trabajo forzado y su uso se remonta a la obra teatral (PLAY) de 1921 del checo Karel Capek titulada R.U.R., Robots Universales de Rossum. En esta obra Capek habla de la deshumanización del hombre en un medio tecnológico; a diferencia de los robots actuales, éstos no eran de origen mecánico, sino más bien creados a través de medios químicos. Pero no fue sino hasta 1942 que el término robótica, es decir el estudio y uso de robots, se utiliza por primera vez por el escritor y científico ruso-americano Isaac Asimov en una pequeña historia titulada Run around. Ahora surge la pregunta:

-¿A partir de cuándo nacen los robots tal y como los conocemos hoy en día?-

Pues fue hacia finales de la década de los 50 y principios de los 60, cuando salen a la luz pública los primeros robots industriales conocidos como Unimates diseñados por George Devol y Joe Engelberger, este último creó Unimation y fue el primero en mercadear estas máquinas, con lo cual se ganó el título de "Padre de la Robótica". Ya en la década de los 80 los brazos industriales modernos incrementaron su capacidad y desempeño a través de microcontroladores y lenguajes de programación más avanzados. Estos avances se lograron gracias a las grandes inversiones de las empresas automovilísticas.



**Figura 2:** Imagen de la película "Metrópolis", 1926

### 1.3.2.1.2. Tipos de robots

Hay varios tipos de robots. Siempre cuando uno piensa en el futuro se imagina una nueva tecnología mucho más sofisticada. La gran variedad que hay de diferentes tipos y funciones programadas de un robot es impresionante. Existen robots de uso doméstico, otros para ayuda médica, otros para labores peligrosas, y bien los robots de la industria. En el punto álgido de los robots más impresionantes están los llamados Androides, es decir robots que semejan características humanas, estos están muy cerca de ser una realidad cada vez más cercana en nuestro mundo.

### 1.3.2.1.3. ¿Qué es un robot?

Existen muchas formas de definir el término robot, podemos utilizar la definición de 1979 del “Robot Institute of America”, que dice:

*“Es un manipulador reprogramable y multifuncional diseñado para mover material, piezas, herramientas o bien dispositivos especializados para desempeñar una variedad de labores a través de movimientos diversos programados.”*

Obviamente esta definición es bastante rígida e insípida, de una forma más personal se puede decir que un robot es un dispositivo que permite realizar labores mecánicas de una manera mucho más eficiente que un humano, y sin necesidad de poner en riesgo la vida de éste.

### 1.3.2.1.4. Robótica en la actualidad

Cada vez son mayores las investigaciones que se inician en temas de robótica, siendo cada vez mayor el número de laboratorios creados y empresas montadas dedicadas a este fin.

El mayor campo de investigación es sin duda la inteligencia artificial, su propósito es lograr hacer “pensar” a una máquina que está compuesta por partes mecánicas y electrónicas, tal y como lo haría un ser humano, pudiendo decidir, tomar decisiones y teniendo la capacidad de aprender. A la cabeza de estas investigaciones están Japón y Estados Unidos, donde se han creado prototipos capaces de entablar conversaciones con un humano.

Habrá que esperar al futuro para ver hasta donde se logra llegar.



**Figura 3:** Androide ficticio

### 1.3.2.2. Visión Artificial

La visión artificial, también conocida como visión por computador o visión técnica, es un sub-campo de la inteligencia artificial.

El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen.

Para un ordenador, una imagen no es una imagen, si pensamos en ella tal y como la concibe un ser humano. Es decir, como un cuadro en el que hay una mezcla de colores de una forma definida que represente algo o refleje algo. En el interior de un computador, esto no es así, es una matriz, del tamaño de la imagen, dividida en puntos, llamados píxeles, que tiene varios valores numéricos cada uno, es decir, una matriz inmensa de números.

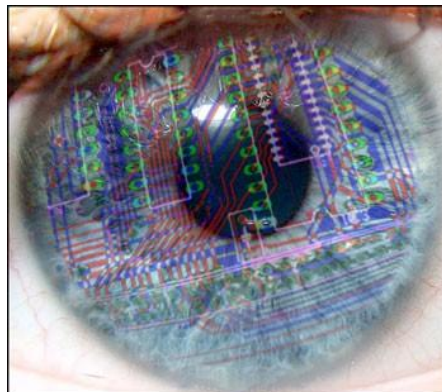
Los objetivos típicos de la visión artificial incluyen:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).
- La evaluación de los resultados (por ejemplo, segmentación, registro).
- Registro de diferentes imágenes de una misma escena u objeto, es decir, hacer concordar un mismo objeto en diversas imágenes.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena; este modelo podría ser usado por un robot para navegar por la escena.
- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.

Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de grafos y otros campos. La visión artificial cognitiva está muy relacionada con la psicología cognitiva y la computación biológica.

En general, la visión artificial se usa para procesos en los que usarla funcionaría mejor que un humano debido a que no le afectan factores de naturaleza humana, tales como cansancio, necesidad de parpadear, necesidades biológicas, etc.

Y en donde se busca ver más allá de lo que se ve. Que el software de captación de imágenes más el procesamiento que se está haciendo sea capaz de sacar más información del entorno, que un humano en la misma situación.



**Figura 4:** Ojo digital

### **1.3.2.2.1. Alcance de la visión artificial**

La visión artificial está limitada por dos factores principalmente.

Es obvio que contra más información tenga la imagen, más se podrá sacar de ella con el procedimiento adecuado. Si el hardware de captación de imágenes que estamos usando es limitado, y la imagen que almacena no tiene la suficiente información para poder ser tratada, vulgarmente dicho, es lo mismo hacerlo que no hacerlo. No nos vale de nada llevar a cabo el procesamiento, puesto que no se sacará ninguna información. La misma persona observando la escena ella misma puede sacar la misma o mejores conclusiones que el software.

Por lo tanto, dependiendo del alcance que queramos tener, es decir, cuanto más queramos saber de la imagen, mejor deberá de ser el hardware de captación, para que así se pueda trabajar y saber que se pueden obtener buenos resultados.

El otro factor viene lógicamente acarreado por el primero.

Si la imagen que queremos procesar, es bastante grande, contiene muchísimos píxeles, habrá que realizar una ingente cantidad de operaciones sobre ellos. También influye el procesamiento que queramos llevar a cabo, pero dado un procesamiento a realizar sobre la imagen, el hardware que tengamos como procesador, debe de tener la suficiente potencia para realizar todos los cálculos, estamos hablando de millones de operaciones en un poco espacio de tiempo. Si el hardware es limitado, es posible que tarde bastante en procesar.

Otro factor más añadido, es la calidad del software realizado. No es sencillo diseñar un buen programa para procesar imágenes.

El principal problema con el que se encuentran los programadores es la memoria. Para realizar todas las operaciones, el programa irá creando buffers, variables, imágenes auxiliares, que van ocupando espacio. Si no se libera o se va borrando a la vez que se va procesando, la memoria llega a acabarse, con lo que el programa se obstruirá y dejará de funcionar. Si esto ocurre en un sistema operativo, es decir, que no es un programa aislado en un microprocesador, el sistema operativo parará el programa antes de que ocurra el error.

Aparte de la memoria, sobra decir, que para obtener la información que se quiere obtener de una imagen, el software diseñado debe de estar programado para realizar las operaciones correctas para llegar a ese fin. En caso contrario puede que nos diese resultados pero que no tienen que ver con lo que queramos saber nosotros de la imagen o la secuencia de imágenes.

### 1.3.3. Estado del arte

Hoy en día estos campos están en completo auge.

En Japón se están desarrollando sistemas de visión artificial que son capaces de reconocer una persona en una estación de tren, o prototipos de humanoides que son capaces de interactuar con una persona humana, aprender e incluso recordar quién eres y tu nombre.

Las máquinas usadas para la fabricación en la industria son cada vez más eficientes y más precisas en su trabajo.

Respecto al objeto de este proyecto, existen muchos prototipos por el mundo, unos son más avanzados que otros.

Pocos se han encontrado con las mismas características que el aquí desarrollado, siendo el dato más curioso que el más parecido, se ha desarrollado en España, más concretamente por un equipo de ingenieros de Albacete en el año 2009. Tal equipo, afirma que su robot es vigilante también, pudiendo ser controlado por el usuario. En un vídeo colgado en la red se ve un poco el funcionamiento que tiene. Dicen que dicho robot va captando imágenes, pero la diferencia, es que luego es el ingeniero quien ve la información recopilada y hace una evaluación.

En el presente proyecto se pretende que sea el mismo vehículo el que tome las decisiones. Un punto fuerte es que es él mismo, el que realiza el procesamiento de las imágenes y de toda la información, no depende de una computadora más potente.

El proyecto español es lo más parecido que se ha encontrado por la red en cuestión de vehículos de vigilancia.

A continuación se muestran varios prototipos encontrados, no con las mismas características que el que se pretende diseñar pero aún bastante convincentes:



**Figura 5:** Diversos Robots de vigilancia



## 1.4. Normas y referencias

### 1.4.1. Normas

Las normas aquí expuestas fueron consultadas y seguidas estrictamente en la medida de lo posible.

UNE157001:2002 “*Criterios generales para la elaboración de proyectos*”

UNE1027 “*Dibujos Técnicos, Plegado de planos*”

### 1.4.2. Bibliografía

- [1] Brian W. Kernighan, Dennis M. Ritchie, *El lenguaje de programación C*, (1991).
- [2] Herbert Schildt, *C - Guía De Autoenseñanza*, (2001).
- [3] Herbert Schildt, *C++ - Guía De Autoenseñanza*, (1995).
- [4] Scott Meyers, *Effective C++*, (1997).
- [5] Manuel Alfonseca, Alfonso Alcalá, *Programación orientada a objetos, Teoría y técnicas OOP para desarrollo de software*, (1992).
- [6] Herb Sutter, Andrei Alexandrescu, *C++ Coding Standards: 101 Rules, Guidelines, and Best Practices*, (2004).
- [7] Gari Bradski, Adrian Kaehler, *Learning OpenCV*, (2008).
- [8] Roberto Rodríguez Morales, Juan Humerto Sossa Azuela, *Procesamiento y Análisis Digital de IMÁGENES*, (2001).
- [9] Neil Reid , Ron Seide, *802.11 (Wi-Fi)*, (2003).
- [10] Erik Cuevas, *Procesamiento digital de imágenes con MatLab Y Simulink*, (2010).
- [11] Rulph Chassaing, *Digital Signal Processing and Applications with the C6713 and C6416 DSK*, (2005).
- [12] NXP B.V., *UM10139 LPC214x User Manual*, (2010).
- [13] BeagleBoard, *BeagleBoard-xM Rev C System Reference Manual Revision 1.0*, (2010).

### **1.4.3. Recursos empleados**

#### **1.4.3.1. Información de internet**

- Información completa de QT en español con un foro bastante activo

<http://www.zonaqt.com>

- Información sobre la comunicación i2c de la placa BeagleBoard-xM

[http://elinux.org/Interfacing with I2C Devices](http://elinux.org/Interfacing_with_I2C_Devices)

- Foro en castellano de electrónica, con integrantes muy capacitados

<http://www.forosdeelectronica.com/>

- Información completa de la biblioteca OpenCV

<http://opencv.willowgarage.com/wiki/>

- Información de las librerías gstreamer

<http://gstreamer.freedesktop.org/>

---

### 1.4.3.2. Software utilizado

---



#### Crossworks for ARM 2.0

Entorno de desarrollo integrado para todos los núcleos ARM comercializado por Rowley Associates. Incluye entre otras características, editor de texto, un intuitivo asistente de proyectos, programación FLASH integrada, y depurador que proporciona las funcionalidades propias del JTAG. Sus prestaciones las comprobamos realizando, depurando, y probando el programa que ejecuta el microcontrolador

---



#### Code::Blocks

Code::Blocks es un entorno de desarrollo gratuito para el lenguaje C++. Es compatible con más de 20 compiladores distintos, incluidos GCC, Microsoft Visual C++, Tiny C, Digital Mars y Borland C++.

---



#### Eclipse

Eclipse es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

---

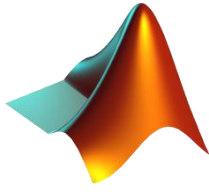


#### CadSoft Eagle 6.1.0

CadSoft EAGLE es un sencillo programa que te permite el desarrollo de circuitos impresos.

El programa consta de tres módulos, un Diagramador, un Editor de esquemas y un Autorouter que están integrados por lo no hay necesidad de convertir los netlist entre esquemas y diseños. Es una potente aplicación con la que diseñar circuitos impresos y realizar esquemas electrónicos. Eagle son las siglas de Easily Aplicable Graphical Layout Editor.

---



### MatLab

MATLAB es un entorno de programación para el desarrollo de algoritmos, el análisis de datos, la visualización y el cálculo numérico. Con MATLAB, puede solucionar problemas de cálculo técnico más rápidamente que con lenguajes de programación tradicionales tales como C, C++ o Fortran.



### Code Composer Studio 5.0

Sistema de desarrollo integrado para desarrollar aplicaciones en procesadores embebidos de Texas Instruments. Code Composer Studio incluye un sistema que opera en tiempo real llamado DSP/BIOS o SYS/BIOS.



### Qt Creator

Qt Creator es un excelente IDE para desarrollar aplicaciones visuales en C++ de manera sencilla y rápida. Como su nombre lo indica, está basado en la librería Qt.



### Dev-C++

Dev-C++ es un entorno de desarrollo para los lenguajes C y C++. Además del consabido editor, añade un compilador y otras utilidades menos evidentes que lo convierten en un gran aliado en la programación.



### Microsoft Visio

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.



### **Microsoft Word**

Microsoft Word es un software destinado al procesamiento de textos.

---



### **VCarve**

VCarve Pro es una solución completa para la CNC de enrutamiento, la rotulación, la talla y el grabado. Es un potente y flexible paquete de software industrial que incluye toda la funcionalidad del diseño, el diseño y mecanizado exigidos por tiendas comerciales y los usuarios, sin dejar de ser muy fácil de usar ya un precio accesible.

---



### **Visual studio 2010**

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

---

### **Mach 2 mill**

Software utilizado para manejar la máquina de control numérico del laboratorio de robótica.

---



### **Calculadoras de Windows y Linux**

La simple calculadora que todo el mundo usa pero que nunca se dice que se ha usado ni se refleja en ningún sitio. Es hora de romper una lanza en su favor.

---

## 1.5. Definiciones y abreviaturas

<b>LPC</b>	Familia de microcontroladores fabricados por NXP, basados en arquitectura ARM.
<b>ARM</b>	Advanced RISC Machine, es una arquitectura de procesadores de 32 bits desarrollada por ARM Limited, que es ampliamente empleada en sistemas integrados. Por sus características de ahorro energético, las CPU ARM dominan en el mercado de dispositivos móviles, donde el bajo consumo de energía es un objetivo crítico de sus diseños.
<b>DSP</b>	Procesador de señales digitales (Digital Signal Processor)
<b>SHARP</b>	Fabricante electrónico, cuyos sensores han obtenido el nombre de la marca
<b>JTAG</b>	Definido por el estándar IEEE Std.-1149.1, es un método para testear circuitos impresos implementados en circuitos integrados.
<b>I2C</b>	Tipo de bus de comunicación que usa únicamente dos cables. Se establece una transmisión de datos maestro esclavo.
<b>SPI</b>	Tipo de bus de comunicación serie.
<b>GND</b>	Masa.
<b>ARM</b>	Arquitectura de microprocesadores.
<b>LED</b>	Diodo emisor de luz (Led Emitter Diode)
<b>PCB</b>	Placa de circuito impreso (Printed Circuit Board)
<b>IDE</b>	Entorno de desarrollo integrado (Integrated design environment)
<b>FPS</b>	Imágenes que capta por segundo una cámara (Frames per second)
<b>Servo</b>	Abreviatura de servomotor.
<b>LiPo</b>	Nombre común que se le da a una batería de tipo Litio-Polímero
<b>AWG</b>	Ancho de los cables en el sistema americano (American Wire Gauge)
<b>HXT</b>	Tipo de conector
<b>GPS</b>	Sistema de posicionamiento global
<b>VLIW</b>	Del inglés (Very Long Instruction Word), tipo de arquitectura de CPU
<b>FIFO</b>	Memoria en la que el primer dato entrante, es el que sale (First In, First Out)
<b>Wi-Fi</b>	Mecanismo de conexión de dispositivos de forma inalámbrica

<b>USB</b>	Bus universal de comunicación serie ( Universal Serial Bus )
<b>DNS</b>	Base de datos, enlaza nombres con IPs ( Domain Name Service )
<b>UART</b>	Bus de comunicación serie y asíncrona (Universal Asynchronous Receiver and Transmitter)
<b>Hat o POV</b>	En un joystick, el hat, conocido también como POV procedente de Point Of View, es un dispositivo pensado para un entorno virtual, normalmente de juego en donde manejas por ejemplo, un avión, el mirar hacia distintos sitios cambiar su trayectoria. Si fuese un personaje, permite lo que sería mirar con la cabeza a distintos sitios, pero sin mover el personaje en sí. Es una pequeña palanca situada normalmente en el extremo superior de un joystick.

## 1.6. Requisitos de diseño

Se pretende diseñar un robot con las siguientes características:

- El robot ha de ser móvil, pudiendo desplazarse.
- Ha de poder ser controlado en todo momento por un usuario.
- Dispondrá de diversos sensores a su alrededor con los que poder comprobar la distancia a la que se encuentre de algún objeto.
- Deberá de tener un dispositivo de captación de imágenes situado en una articulación, que le permita enfocar en diferentes direcciones, captando imágenes en tiempo real. El hardware al que esté conectado este dispositivo debe ser lo suficientemente potente, para llevar a cabo un procesamiento de las imágenes captadas, a la vez que transmitiendo estas a un terminal electrónico permitiendo a un usuario ver dichas imágenes.
- El vehículo dispondrá de dos modos de funcionamiento, uno autónomo y otro controlado por el usuario. En el modo controlado por el usuario, es este último el que gobierna al coche a través de un dispositivo conectado a él. Tendrá control sobre el movimiento del robot, así como de la articulación citada anteriormente. En el modo autónomo es el vehículo el que se gobierna a sí mismo. Aquí debe de ser capaz de evitar un obstáculo en su trayectoria y detectar algún elemento en las imágenes previamente designado, avisando de que ha sido hallado. En este último modo se entrará cuando el usuario lo indique.



**Figura 6:** Esquema general de la idea



- La comunicación se hará de forma inalámbrica a través de una red Wi-Fi privada.
- Estará alimentado por una batería que le proporcione una autonomía de al menos una hora. Dispondrá a su vez de una fuente de un circuito que distribuya la energía a todo el vehículo. El nivel de la batería debe ser conocido por el usuario en todo momento. Contará con un interruptor de encendido y apagado manual, a su vez que protecciones para cortocircuitos o sobrecargas.

## **1.7. Análisis de soluciones**

Durante este apartado se estudian las diferentes estrategias para cumplir con los requisitos de diseño, elección de los componentes y línea a seguir de programación.

### **1.7.1. Hardware**

A continuación se describen todas las placas disponibles en el proyecto, así como sensores y dispositivos para usar en el. Además de cómo montar el vehículo.

#### **1.7.1.1. Placas de desarrollo usadas**

El llevar a cabo toma de decisiones y ejecutar tareas de movimiento lleva consigo una larga programación pero lo más importante un dispositivo capaz de poder ejecutarla. Debe de tener la suficiente potencia para todo el procesamiento que se pretende realizar, para ello se han tenido en cuenta varias placas, se describen a continuación.

Primero se pensó en utilizar únicamente la placa desarrollada en la asignatura “Aplicaciones de sistemas y automática”, esta placa dispone de un microcontrolador modelo LPC2148.

Está fabricado por la casa NXP, basado en arquitectura ARM, más concretamente ARM7. El conocimiento de este microcontrolador se conoce ya de antemano debido a la experiencia de haber trabajado en las prácticas de la asignatura “Informática Industrial” con el microcontrolador modelo LPC2378, un microcontrolador mucho más potente que el anterior pero de igual programación, debido a que pertenece a la misma familia, la LPC2000.

Volviendo al 2148, este dispone de pines de entrada y salida programables, generador de PWM, comunicación I2C, dos temporizadores, conversores analógico/digital y otros periféricos. Estos funcionan a 15MHz. Es la perfecta opción para manejar todos los dispositivos que afectasen al movimiento del coche y a la lectura de sensores.

Para cumplir el objetivo de procesar imágenes y establecer una comunicación inalámbrica con otro dispositivo, no es suficiente, harían falta módulos de memoria y otros elementos externos, para facilitar al microcontrolador trabajar con tal cantidad de datos.

Teniendo en cuenta también el tiempo que le llevaría todo este trabajo..



**Figura 7:** Placa de desarrollo basada en LPC2148

Viendo la dificultad que presentaba utilizar únicamente esta placa, hubo que pensar en otra alternativa. La idea que vino a continuación fue la de no usar una, sino dos placas. Se pensó que se podría usar el microcontrolador, anteriormente descrito, para realizar las tareas que afectasen al movimiento del robot y a la lectura de los sensores. El hardware adicional sería más potente para que su tarea fuese el llevar a cabo el tratamiento de las imágenes, las comunicaciones y la toma de decisiones.

Ambas debían de comunicarse una con la otra de alguna manera, existían principalmente dos modos:

- I2C
- SPI

La comunicación SPI es más rápida que la I2C.

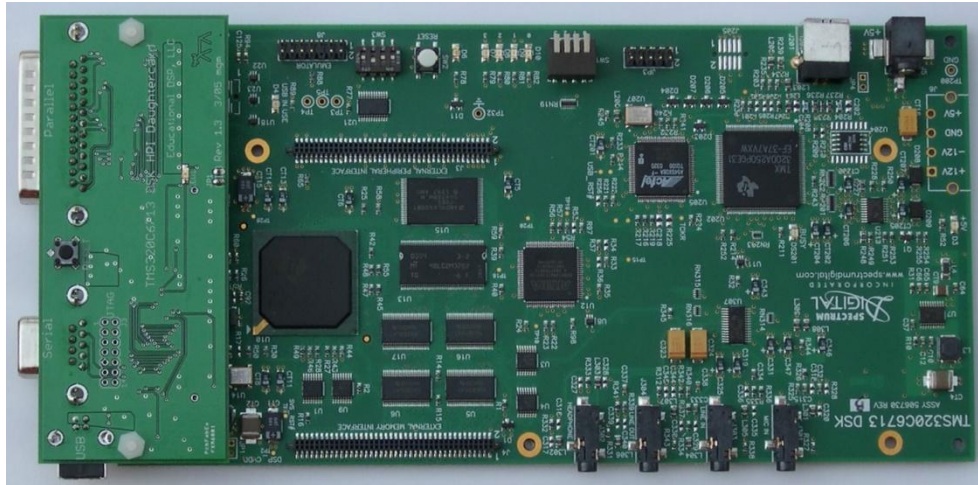
Un bus SPI dispone de 3 conexiones de comunicación comunes, más una conexión más por cada dispositivo conectado al mismo bus, esta conexión adicional es directa al dispositivo e independiente de las demás.

El funcionamiento es muy sencillo: cuando se desee comunicar con un dispositivo, se activará la conexión directa con el destinatario, a continuación, se transmite la información al bus a través de las líneas comunes, el dispositivo que tenga activada su propia conexión será el que reciba la información.

En la comunicación I2C, se utilizarán únicamente dos líneas comunes, a ellas se conectarán todos los dispositivos a los que se desee transmitir información. El funcionamiento es un poco más complicado que el I2C, pero no dejar de ser relativamente sencillo.

Para su funcionamiento, se establecerá un maestro, que será quien maneje el bus, todos los demás dispositivos serán esclavos, que estarán identificados por una dirección. Cuando se quiera transmitir información a un esclavo, se mandará al bus la dirección, si el esclavo está activo escuchando el bus, este mandará una señal, diciendo que está preparado para recibir la información, solo resta mandar la información al bus que será recogida por el esclavo en cuestión. El hecho de tener que identificar el destino de la información, antes de poder transmitirla, se traduce en un tiempo extra, lo cual ralentiza la comunicación.

Pensando en la placa que poner en conjunto con el microcontrolador, se pensó en utilizar un DSP, ya que se disponía de uno en el laboratorio. La placa pertenece a un kit de desarrollo fabricado por Spectrum Digital. Incorporaba en sí un DSP de la casa Texas Instruments, más concretamente el modelo TMS350C6713.



**Figura 8:** Placa DSP TMS350C6713

Con una velocidad de 225 MHz, con arquitectura VLIW, tenía capacidad para llevar a cabo el procesamiento de imágenes y la toma de decisiones, disponía a su vez de comunicación SPI e I2C, lo que no supondría un problema para comunicarla.

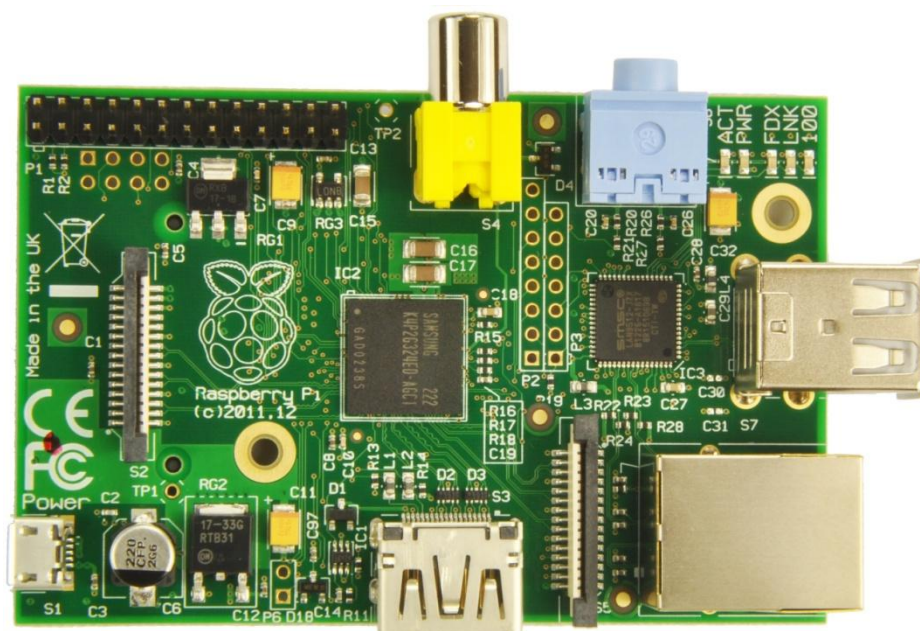
Tras empezar a trabajar con ella, se observaban problemas a solucionar, principalmente dos:

- Determinar la forma pasarle las imágenes. Dispone de un puerto de interfaz de memoria.
- Conectarla a una red Wi-Fi.

Más adelante, se consideró cambiar el DSP por otra placa de propósito general, más potente, que incluyese un sistema operativo, el cual facilitaría el trabajo.

En el mercado existen varias placas de este tipo. Un primer modelo podría ser la placa denominada Raspberry Pi. Una placa computadora de bajo coste desarrollada en Reino Unido.

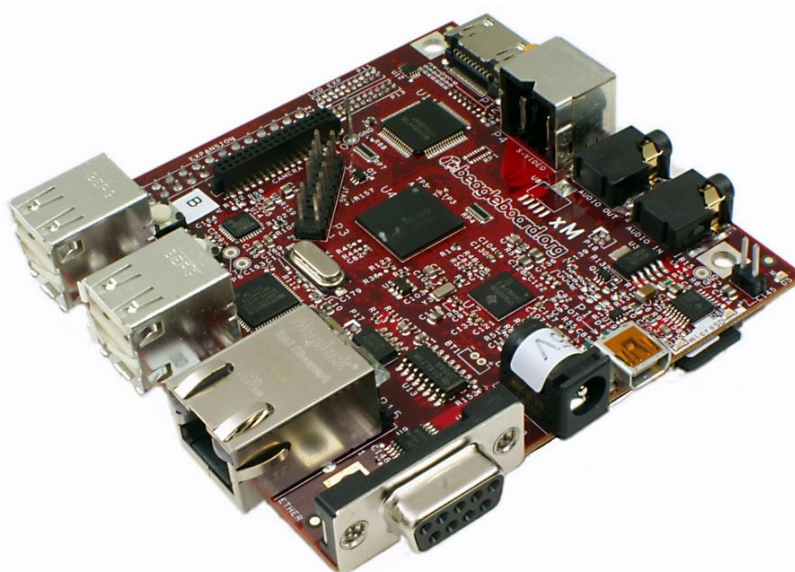
El diseño incluye un System-on-a-chip Broadcom BCM2835 que contiene un procesador ARM1176JZF-S a 700 MHz y 256 MB de memoria RAM, desde donde se puede ejecutar LINUX. Se usará una tarjeta SD como disco duro. Es decir, un PC Embebido con sistema operativo Linux. Las ventajas de un sistema operativo como Linux son notables.



**Figura 9:** Raspberry Pi

Se encontró información de otra placa denominada BeagleBoard-xM. Basada en ARM Cortex A8, a 1 GHz. También dispone de una tarjeta SD como disco duro. Frente a la Raspberry, esta disponía de 4 puertos USB, conexión Ethernet, conexión RS232, comunicación I2C, entrada y salida de audio y un punto bastante fuerte a su favor, dispone de una puerto paralelo para conectar una cámara cuya conexión sea de este tipo.

La ventaja más evidente era la mayor velocidad que poseía. En ella también se ejecuta Linux, con lo que con esta placa también dispondríamos de un sistema operativo completo.



**Figura 10:** BeagleBoard-xM



### 1.7.1.2. Movimiento

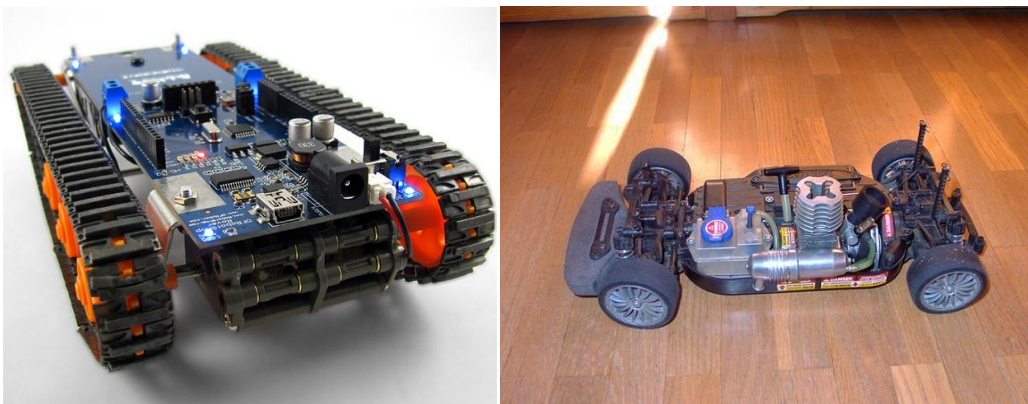
Una de los factores más elementales era decidir cómo se iba a mover el coche, habiendo dos posibilidades: tipo tanque o tipo coche.

- Tipo tanque:
  - Movimiento más preciso
  - Control de dos motores
  - No es necesario servomotor de dirección
  - Movimiento no continuo.
  - Más tiempo en corregir rumbo.
  - Dispone de 4 ruedas, ambas de dirección y de tracción o en su lugar 2 orugas.
  
- Tipo coche:
  - Movimiento continuo.
  - Un solo motor
  - Servomotor de dirección
  - Difícil maniobrabilidad
  - Movimiento más rápido
  - Dispone de 4 ruedas, 2 dirección, 2 tracción.

Las características podrían señalar que la mejor forma sería un movimiento tipo tanque desde el punto de vista de que es mejor para que se gobierne así mismo el vehículo, además es sabido la gran cantidad de robots y microrrobots usan este movimiento. Pero no se debe desestimar del todo el movimiento tipo coche.

Funciones tales como enderezar, seguir o rectificar la dirección supondrían una tarea más dura de trabajo, ya que se pretende que el coche esté en continuo movimiento sin tener que detenerse para girar, salvo en situaciones concretas.

En la figura 9 se muestra un ejemplo de cada uno de estos tipos, a la izquierda vemos el tipo tanque y a la derecha el tipo coche.



**Figura 11:** Tipos de configuraciones según movimiento

### 1.7.1.3. Motor y controlador

Al robot le hace falta un motor de pequeño tamaño pero que sea capaz de moverlo, que tenga cierta fuerza. El motor que se considera es un modelo RS-360H. Cuyas características vienen enumeradas más adelante. Aunque es un modelo de motor antiguo, el funcionamiento se ha probado y se tiene experiencia con él, comprobando que funciona correctamente. Aparte se disponía de uno.

Se tiene en cuenta usar uno diferente, aunque el funcionamiento del otro es correcto, usar uno nuevo también supone trabajo de carrocería, ya que el que se posee viene incrustado en un coche. Además de suponer un coste económico el adquirirlo.

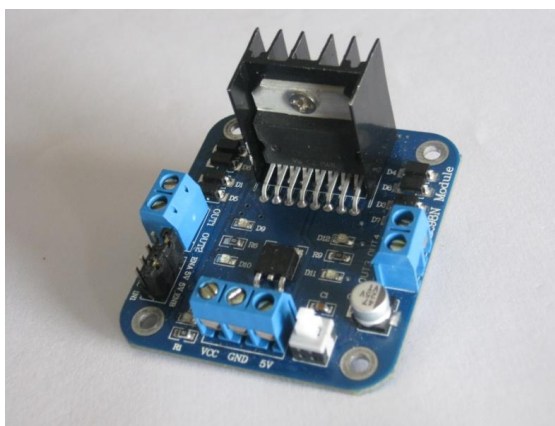
Para controlarlo, se puede implementar un pequeño circuito con transistores y puertas lógicas, lo que se denomina un puente en H. Este había de ser capaz de soportar la intensidad del motor que se eligiese, añadiendo trabajo de diseño y fabricación de la placa.

También se conocía el integrado L298, un doble puente en H, capaz de manejar dos motores de continua, a una tensión de hasta 50 voltios y de hasta dos amperios por motor. Bastante robusto para nuestra aplicación, solo se usaría de él un puente, al estar incluido todo en un circuito integrado reducía el trabajo de diseño.



**Figura 12:** Integrado L298

Para usarlo era necesaria una pequeña electrónica a su alrededor. Pero existen unos módulos que se comercializan en donde ya viene todo montado, ahorrando trabajo a la vez que dinero, ya que el coste de la placa, era el mismo que comprar únicamente el integrado.



**Figura 13:** Módulo L298

#### 1.7.1.4. Estructura

Hay que montar todo el sistema en alguna plataforma, que debe de ser robusta y aguantar todo el peso de cada componente que se quiera usar, para ello se consideran dos opciones:

- Una es construir todo desde cero, usando madera u otros materiales, esto supone un trabajo de diseño y fabricación. Habría que, una vez decidida la forma del coche, empezar a fabricar todo, teniendo que adquirir, ruedas, ejes, motor, lo que aparte de trabajo, supondría un coste económico.
- Otra es usar de base un coche teledirigido antiguo, este disponía de carcasa, las ruedas estaban en buen estado y se podría aprovechar parte de su electrónica. Una ventaja es que se podría utilizar el mismo motor que trajese el coche teledirigido original. Además se le puede dejar la carcasa e incluirle la electrónica bajo esta, esto haría que las placas tuviesen una protección, estando cubiertas. Además, de que el ser tipo Pick-Up, permitía que la articulación que se pide, este en una zona adecuada.



**Figura 14:** Coche teledirigido original

Eligiese lo que se eligiese, el chasis del coche, había de ser bastante robusto, para soportar todo el peso que conllevase la electrónica, servomotores, sensores y sobre todo la batería.

### 1.7.1.5. Cabeza articulada

La cabeza articulada es donde va instalada la cámara, se pretende que la cámara no vaya sola, sino que se acompañe de sensores para fortalecer el procesamiento de imágenes con señales captadas al mismo tiempo de distancia.



**Figura 15:** Ejemplo de cabeza articulada

Una primera idea es la de implementar en la misma vertical, la cámara con una articulación, y un radar giratorio 360° girando constantemente.

Esto traería problemas debido a la conexión de un elemento giratorio constante. Aunque se sabía la forma de llevarlo a cabo, era tediosa y se descartó.

Otra forma de hacer esto es implementar el radar en la misma dirección en la que mira la cámara, así, a lo que se esté apuntando, u observando, se puede saber a qué distancia está.

Una solución más, difícil pero no descartable, es usar la estéreo visión, colocar dos cámara en paralelo, con un procesamiento se podría estimar la distancia a la que estuviese el objeto enfocado con estas, funcionando como lo hacen los ojos humanos.

### 1.7.1.6. Cámara

La cámara para instalar en el vehículo debía de ser una cámara con buena resolución. A parte de proporcionar imágenes que se pudiesen procesar sin problemas. Intentando evitar el tener que transformarlas a otro formato antes de ser procesada, debido a que esto consume gran parte de recursos. Debía de ser de fácil configuración y manejo. La primera cámara que se adquirió fue un módulo basado en el sensor de imagen mt9m111. Un módulo capaz de captar imágenes a 54Mhz, programable por i2c y conexión por un puerto paralelo de 8bits.





**Figura 16:** Primera cámara usada

El que dé las imágenes directamente por puerto paralelo significa más velocidad para recibirlas, pero tiene un inconveniente, el módulo y la placa en la que va conectada no pueden tener mucha distancia entre ellas, debido a efectos de transmisión de señal, con lo cual, el poder ponerlo en un sitio elevado junto con la articulación se dificulta.

La otra opción, es utilizar una webcam, claro, tiene el inconveniente de que la placa a la que se conectase, la placa que se elija finalmente, debe de tener puerto USB y poder implementar los drivers para el manejo de una cámara de estas características. Aunque está la ventaja de que hay una amplia variedad de cámaras en el mercado. Una de ellas, se conoce en particular, debido a la experiencia de personas que la han usado para distintos propósitos, el modelo en cuestión es la cámara “Logitech c210”.



**Figura 17:** Webcam Logitech c210

La ventaja de usar una webcam, es que respecto a la cámara por puerto paralelo, es la comodidad de la conexión, además de permitir más distancia entre la cámara y la placa, debido a que el cable USB era de 1m de largo.

Con esta cámara, a las imágenes recogidas se les debía hacer una transformación para poder ser trabajadas en OpenCV.

### 1.7.1.7. Articulación

La cámara debía de estar articulada, para cuando se desee poder mirar hacia donde se le indicase. Había dos formas principales de montar una articulación.

- Utilizando dos servomotores, uno para el movimiento horizontal y otro para el vertical. El inconveniente está en que un servomotor solo gira 180°, con lo que solo se iba a poder recorrer un cuarto de circunferencia. Pero su control es muy sencillo, teniéndose experiencia personal trabajando con ellos.  
Los soportes para crear esta articulación junto con dos servomotores, se comercializan y es algo bastante habitual en tiendas de robótica. Siendo más fácil dado a los tamaños estándares de los servomotores.
- Utilizar motores paso a paso, con estos motores, se podría crear una articulación que si girase 360°. Con estos se podría conocer bastante bien el estado en el que estuviese la articulación y moverla con precisión. Es más difícil su manejo, pero lo peor sería que si se usa estos motores para la articulación, se tendría que diseñar un modo de que la cámara también pudiese girar 360° sin que los cables se dañasen ni se liasen.

### 1.7.1.8. Sensores

Al tener que conocer la distancia a la que se está de los obstáculos se deben de instalar sensores de distancia por todo el vehículo.

Para ello se conocen dos tipos de sensores de distancia, el tipo Sharp y el radar ultrasónico.

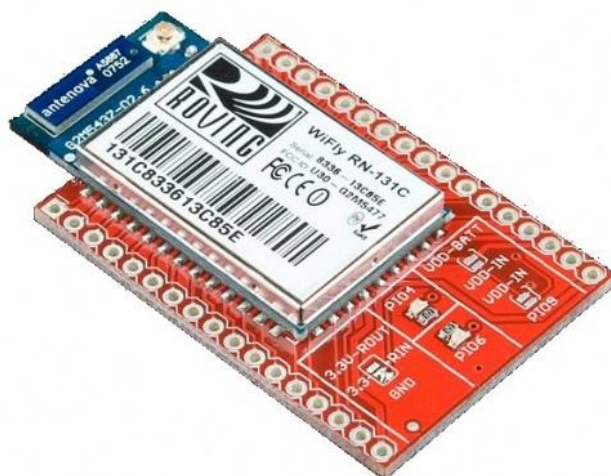
- Tipo Sharp: Este es un sensor de distancia que funciona por infrarrojos, el funcionamiento es bastante sencillo. Consta de un diodo emisor de luz infrarroja y un receptor, en base a la recepción, dará una salida. Funcionan a 5 voltios y su salida es analógica. Son bastante fiables.
- Radar ultrasónico: Este consta de un emisor de ultrasonidos y de un receptor. El funcionamiento se basa en la escucha, comprobando el tiempo que se tarda en recibir el eco provocado por la emisión del ultrasonido. Su inconveniente es que funciona en un cono. Es decir, proporcionará la misma salida, si cogemos un objeto y los desplazásemos por lo que sería un estéreo radián con centro en el sensor.

Estos eran los dos tipos que se estudiaron como sensores de distancia.

### 1.7.1.9. Conexión Wi-Fi

Había dos opciones para llevar a cabo la conexión a una red Wi-Fi. En el mercado existen una gran variedad de dispositivos que se pueden usar para este fin. Pero de todas las posibles han sido dos, las que se han valorado.

La primera es usar un módulo de radio, del fabricante Roving Networks. El dispositivo en cuestión es el modelo Wifly-GSX -RN131. Este es de pequeño tamaño permitiendo el acceso a una red siguiendo el estándar IEEE 802.11b/g. Viene preparado con entradas analógicas, entradas y salidas digitales y una UART. Su funcionamiento es bastante sencillo. Una vez configurado, no tendrá problemas en comunicar información de forma inalámbrica. Para pasarle información de cara a que la envíe a través de Wi-Fi, o recibir la información que le llegue por dicha red, se hará por la UART. Es posible transmitir información por esta comunicación serie a 2.4 Mbps, pero por red inalámbrica solo soporta 1 Mbps. Como se quiere transmitir imágenes, a la vez que información, hay dudas de que este nos pueda ser la elección final.



**Figura 18:** Módulo Wifly-RN131-GSX

La otra opción era que se usase un conector por USB capaz de permitir el acceso a una red Wi-Fi. Hay muchos en el mercado y de diferentes potencias, con tasas de transferencias adecuadas para el objetivo que perseguimos. Para usar uno de este tipo habría que decantarse por usar una placa que tuviese conector USB. Esto implicaría la configuración del dispositivo en la placa. Tarea de la que no sabíamos si iba a resultar complicada o no.

Para la conexión a internet se pensó en utilizar un servicio de D.N.S. el cual cuando se conectaba a internet se registraba la IP con la que entraba con un nombre determinado, desde el ordenador solo había que llamar a ese nombre para conectarse a él.

A parte de la conexión a la red Wi-Fi, se debatió también sobre a qué red conectarse. Una de las opciones más elementales era usar la red misma de la escuela de ingeniería. Pero tenía los siguientes problemas:

Muchos usuarios al mismo tiempo conectados: esto provoca la saturación de la red y la caída de la misma. Aparte de la autenticación como universitario para poder acceder a internet.

La otra opción era de la de crear una red Wi-Fi propia usando un router. Esta red no tendría ocupación salvo la nuestra. Teniendo total disponibilidad para trabajar con ella sin que nadie más tuviese acceso a la misma.

### **1.7.1.10. Alimentación**

Para alimentar el vehículo es obvio que se tenía que usar una batería, la elección debía de hacerse sobre el tipo de batería que emplear, ya diferentes tipos tienen características distintas, el tipo que eligiésemos debía de proporcionar al menos una autonomía de una hora, tal y como indican los requisitos, los tipos de batería estudiados son los siguientes:

#### **1.7.1.10.1. Tipos de baterías**

##### **- Baterías de níquel-cadmio (Ni-Cd)**

Utilizan un cátodo de hidróxido de níquel y un ánodo de un compuesto de cadmio. El electrolito es de hidróxido de potasio. Esta configuración de materiales permite recargar la batería una vez está agotada, para su reutilización. Sin embargo, su densidad de energía es de tan sólo 50 Wh/kg, lo que hace que tengan poca capacidad. Admiten sobrecargas, se pueden seguir cargando cuando ya no admiten más carga, aunque no la almacena. Admiten un gran rango de temperaturas de funcionamiento.

- Voltaje proporcionado: 1,2 V/célula
- Densidad de energía: 50 Wh/Kg
- Capacidad usual: 0,5 a 1,0 A (en pilas tipo AA)
- Efecto memoria: muy alto

##### **- Baterías de níquel-hidruro metálico (Ni-MH)**

Utilizan un ánodo de hidróxido de níquel y un cátodo de una aleación de hidruro metálico. Este tipo de baterías se encuentran menos afectadas por el llamado efecto memoria. No admiten bien el frío extremo, reduciendo drásticamente la potencia eficaz que puede entregar.

- Voltaje proporcionado: 1,2 V/célula
- Densidad de energía: 80 Wh/Kg
- Capacidad usual: 0,5 a 2,8 A (en pilas tipo AA)
- Efecto memoria: bajo

**-Baterías de iones de litio (Li-ion)**

Las baterías de iones de litio (Li-ion) utilizan un ánodo de grafito y un cátodo de óxido de cobalto, trifilina ( $\text{LiFePO}_4$ ) u óxido de manganeso. Su desarrollo es más reciente, y permite llegar a altas densidades de capacidad. No admiten descargas y sufren mucho cuando estas suceden; por lo que suelen llevar acoplada circuitería adicional para conocer el estado de la batería, y evitar así tanto la carga excesiva, como la descarga completa. Apenas sufren el efecto memoria y pueden cargarse sin necesidad de estar descargadas completamente, sin reducción de su vida útil. No admiten bien los cambios de temperatura.

- Voltaje proporcionado:
  - A Plena carga: entre 4,2 V y 4,3 V dependiendo del fabricante
  - A carga nominal: entre 3,6 V y 3,7 V dependiendo del fabricante
  - A baja carga: entre 2,65 V y 2,75 V dependiendo del fabricante (este valor no es un límite, se recomienda)
- Densidad de energía: 115 Wh/Kg
- Capacidad usual: 1,5 a 2,8 A (en pilas tipo AA)
- Efecto memoria: muy bajo

**-Baterías de polímero de litio (LiPo)**

Son una variación de las baterías de iones de litio (Li-ion). Sus características son muy similares, pero permiten una mayor densidad de energía, así como una tasa de descarga bastante superior. Estas baterías tienen un tamaño más reducido respecto a las de otros componentes.

Las baterías LiPo se venden generalmente de 1S a 4S lo que significa:

- Li-PO 1S: una celda, 3,7 V.
- Li-PO 2S: dos celdas, 7,4 V.
- Li-PO 3S: tres celdas, 11,1 V.
- Li-PO 4S: cuatro celdas, 14,8 V.

Cada celda tiene un voltaje nominal de 3,7 V, Voltaje máximo 4,2 y mínimo 3,0. Este último debe respetarse rigurosamente ya que la pila se daña irreparablemente a voltajes menores a 3 volts. Se suele establecer la siguiente nomenclatura XSYP que significa X celdas en serie, e Y en paralelo. Por ejemplo 3s2p son 2 baterías en paralelo, donde cada una tiene 3 celdas o células. Esta configuración se consigue conectando ambas baterías con un cable paralelo.

Había que elegir entre uno de estos tipos.

---

## 1.7.2. Software

Dentro de la parte de software, hay que distinguir tres partes:

- Microcontrolador
- DSP
- Ordenador

### 1.7.2.1. Software microcontrolador

Se tenía en principio el conocimiento de la programación en lenguaje C, pero debido al avance en el estudio de la programación, se terminaría programando en C++. Éste proporciona grandes avances y ventajas respecto al lenguaje C.

Para programarlo, se usaría el software que se ha utilizado siempre para ello, el IDE CrossWorks, utilizando de nuevo el conector JTAG para ello.

### 1.7.2.2. Software DSP

Para el software del dsp que se utilizase, se programaría en C/C++ ya que es el lenguaje conocido. Dependiendo de la placa escogida se utilizaría un programa u otro. Para el procesamiento de imágenes, estaba la posibilidad o de procesarlos uno mismo, buscando las funciones matemáticas indicadas e implementarla, o también la de usar las librerías OpenCV, código abierto divulgado en internet, en el que las funciones que incluye proveen de procesamientos de imágenes completos.

### 1.7.2.3. Software Ordenador

En la parte del ordenador se tenían tres opciones para usar. Una era usar MatLab, aprendido en la asignatura de “Aplicaciones de Sistemas y Automática” reforzado por estudio propio. La cual incluía una potente toolbox de procesamiento de imágenes con documentación suficiente para poder utilizarla.

Una segunda opción era usar Visual Studio 2010, licenciado por la UCA para sus estudiantes, proporcionaba una potente herramienta para el diseño de interfaces gráficas, pero en lenguaje C#, del que no se tiene conocimiento.

La tercera opción era QT, programa con licencia gratuita, dedicado a la creación de interfaces de usuario y que se programa en C++.

## **1.8. Solución adoptada**

A continuación se describe todo el proceso de cómo alcanzar los requisitos que han de cumplirse, siguiendo todo el desarrollo del robot, montaje, conexión, funcionamiento de cada parte, etc.

### **1.8.1. Decisiones tomadas**

En este apartado se detallan las decisiones finales en diversos apartados del sistema, tales como placas electrónicas utilizadas, sensores empleados, software y demás elementos. Se explica porque se han tomado las decisiones y porque se han desestimado las otras.

#### **1.8.1.1. Estructura**

Se montará un vehículo. Este tendrá un soporte elevado en donde irá la cámara que se elijese, teniendo así una zona más elevada de visión. Se usará el coche teledirigido antiguo como base para empezar a construir sobre él, pero se desechará la carcasa, construyendo nuestras propias plataformas para lo que nos hiciese falta.

#### **1.8.1.2. Movimiento**

El movimiento se ha determinado tipo coche, con cuatro ruedas, dos de dirección, las delanteras, y dos de tracción, las traseras, utilizando las que trajese el coche original teledirigido. Es más difícil que el usar tipo tanque pero resulta más fácil el manejarlo.

Al querer desplazarse de forma continua, esta es la manera apropiada de hacerlo, ya que el movimiento tipo tanque, es conveniente detenerse para girar y luego seguir con la marcha. Con este tipo movimiento se puede manejar con más suavidad.

#### **1.8.1.3. Motor**

Se ha obtenido por continuar con el mismo motor que trajese el coche teledirigido, este venía en muy buena posición en el chasis del coche y el trabajar con él ahorraría en costes y en tiempo de trabajo ya que al usar uno nuevo habría que instalarlo.

Es un modelo RS-360H, es un motor de corriente continua, al que le controlaremos la velocidad por tensión, para ello recurriremos a la generación de PWM. Esto se hará a través del circuito integrado L298, que resulta mucho más cómodo que tener que fabricar el propio circuito controlador con transistores. El integrado elegido soporta dos amperios por motor, se le pueden conectar dos, en nuestro caso solo le conectaremos uno. Utilizaremos el módulo que viene ya

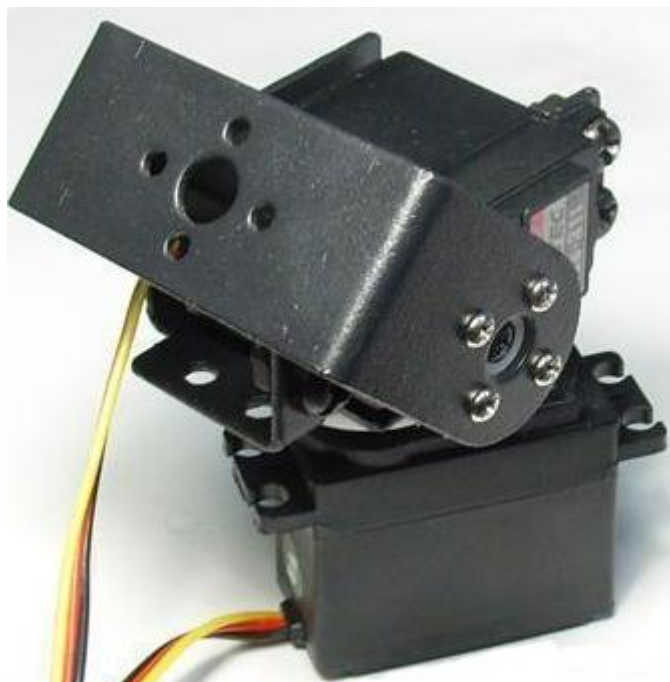
implementado con todo lo necesario para poder manejar el motor, teniendo que conectar únicamente los cables.

#### 1.8.1.4. Electrónica usada

Las placas electrónicas que se van a usar en el vehículo van a ser definitivamente dos, una será la LPC2148, cuyo objetivo será el movimiento del coche y la lectura de los sensores, y la otra será la beagleboard-xM, tras una larga comparativa esta es la que presenta las mejores características para esta aplicación.

#### 1.8.1.5. Cabeza articulada

Este es el nombre que se le ha dado a la parte móvil del vehículo en donde se va a situar la cámara y diferentes sensores. Será móvil y podrá mirar en varias direcciones. Para crearla se ha optado por una articulación compuesta de dos servomotores y brackets de aluminio.



**Figura 19:** Articulación con Servomotores Hitec HS311

Es sencilla de manejar, pero su gran ventaja es la fuerza que tienen los servomotores, del mismo modelo que el usado para la dirección, tienen una fuerza de 3kg/cm. Fuerza de sobra para mover todo lo que nosotros le queremos incorporar a la cabeza.



La cámara que se va a usar va a ser la Logitech c210, puesto que se empleará la beagleboard disponiendo esta de puertos USB. Además dispone de un metro de cable, con lo que se podría colocar la cabeza entera en una posición elevada, que el cable llegaría sin problemas.

### 1.8.1.6. Sensores

Se usaran los dos tipos de sensores discutidos, tanto los infrarrojos, como el radar de ultrasonido, su posición se irá decidiendo a medida que avance el montaje, así como su cantidad.

En el caso de los sensores infrarrojos, se usará el modelo GP2D12, cuyo rango de medida está entre los 10 y los 80cm, distancia suficiente para nuestro robot.

Para el radar, se usará el Parallax Ping)), un sensor de ultrasonidos cuyo rango abarca desde los dos centímetros hasta los tres metros.

### 1.8.1.7. Conexión Wi-Fi

Puesto que usaríamos una placa con puertos USB, en la que se pudiese incorporar un sistema operativo, la mejor solución para una conexión inalámbrica es la de usar un módulo adaptador Wi-Fi por USB, con este elemento se puede usar una alargador USB y colocarlo donde mejor nos convenga sin tener problemas de cableado.

El que se usase debía de valer para la Beagleboard-xM, la elección definitiva se realizó con el siguiente adaptador:



**Figura 20:** Adaptador Wi-Fi USB

### **1.8.1.8. Alimentación**

Respecto a la alimentación se ha decidido que la mejor opción de batería para este robot es una del tipo Litio-Polímero (Li-Po).

No tienen efecto memoria, así que se pueden recargar estén o no a bajo nivel de carga. Proporcionan una gran autonomía, y disponen de conexión para comprobar su estado de carga. Habrá que diseñar una fuente de alimentación para distribuir toda la energía ya que los dispositivos funcionan a diferentes tensiones, además de protecciones para todos los circuitos, para que en el caso de que ocurra un cortocircuito el vehículo quede protegido.

## 1.8.2. Montaje

A continuación se detalla todo el proceso de montaje del vehículo, como se ensambla todo, las piezas utilizadas, cableado y tornillería.

### 1.8.2.1. Base

Para iniciar el montaje del robot, se ha utilizado finalmente el coche teledirigido antiguo. Tenía un buen tamaño, sin ser excesivamente grande, estaba hecho de plástico, material bastante fácil de trabajar.

Para poder empezar a trabajar en él, primero hubo que cortarle bastantes piezas sobrantes. Se eliminó toda la electrónica, lo único que se reservó fue el chasis con las ruedas y el motor.

En la siguiente figura se muestra el coche teledirigido original antes de ser trabajado.



**Figura 21:** Vehículo en su forma original

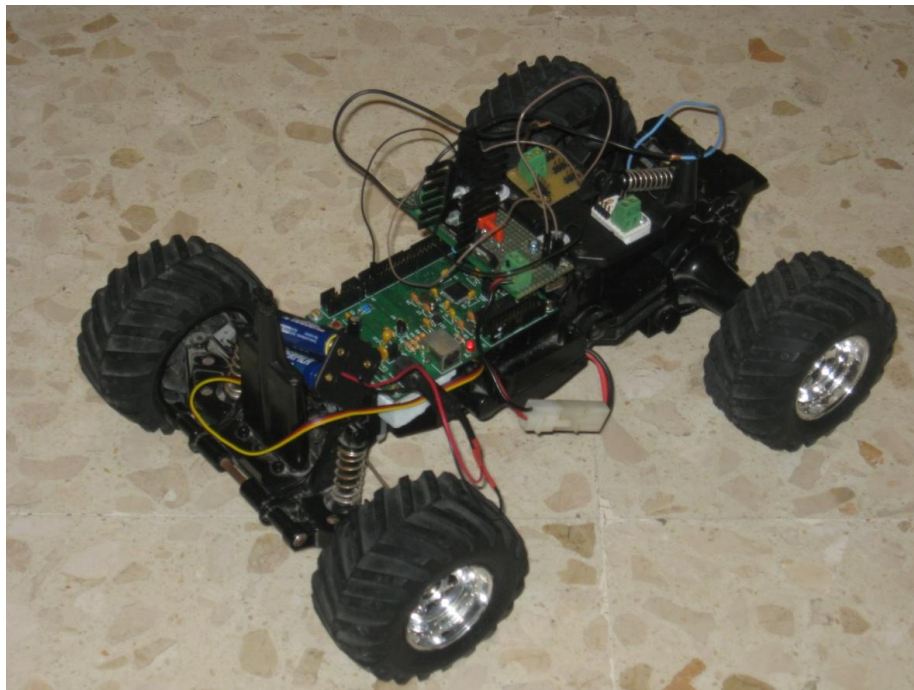
Como se aprecia en la figura, el vehículo simula un coche modelo pick-up a escala. La carrocería que se ve, la carcasa, se quería utilizar en principio, dado que en el interior había bastante espacio para albergar componentes, además de proporcionar una superficie perfecta en el exterior, en donde atornillar sensores y demás elementos.

Un punto fuerte era lo que sería la zona de carga. Donde se podría implementar cualquier zona elevada para la articulación que se hiciese. Luego se desestimó debido al aspecto que mantenía y a que el espacio interno bajo la carcasa no era suficiente.

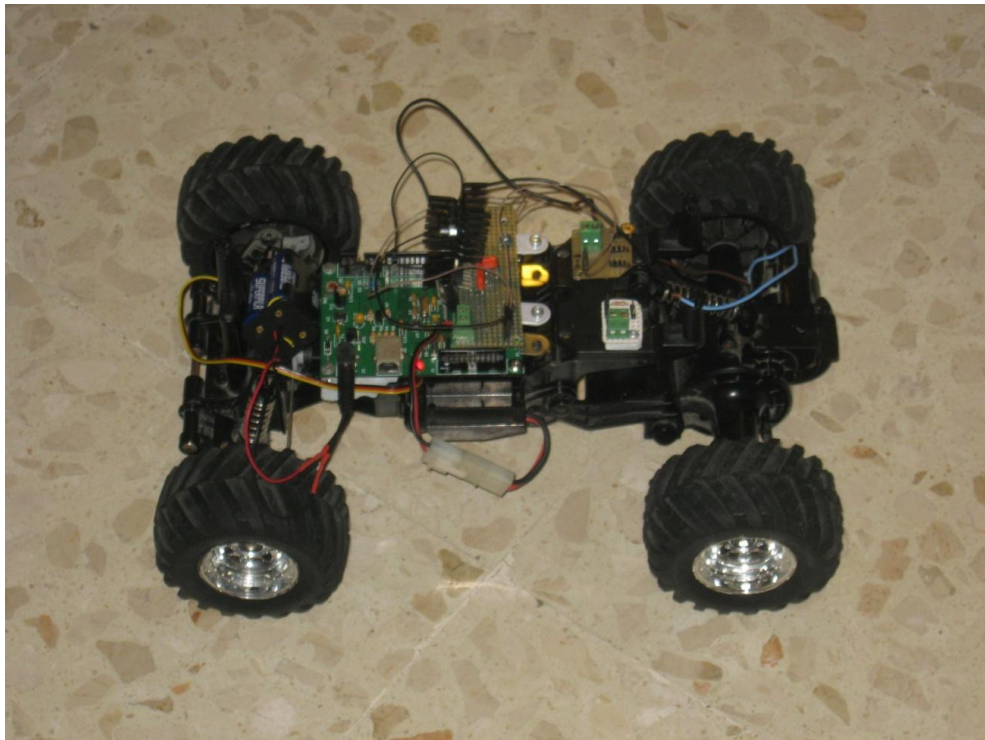
Se ve en la imagen, a través del hueco de la luna delantera la circuitería electrónica que posee, bastante antigua, aún operativa, pero desechable. Se pretendía implementarla toda por completo. A parte de esta placa, se quitó también el servo de dirección que poseía de fábrica, un servo de 6 cables, al que le hacía falta un circuito externo para su control. Fue retirado también.

Una vez eliminadas estas cosas se procedió a trabajo de artesanía y se empezó a recortar, alisar y preparar toda la superficie. Hay que decir que este vehículo se ha ido diseñando a la vez que se ha ido construyendo. No se ha partido de una idea original, puesto que en principio se quería usar la carcasa negra, que se ha explicado el por qué se desestimó.

Cuando el vehículo estuvo limpio lo único que teníamos era el chasis, con las ruedas, el motor y los cables del motor. Esta iba a ser la base con la que nosotros trabajaríamos. Ya se utilizó por primera vez en la asignatura Aplicaciones de Sistemas y Automática para un primer trabajo como se muestra a continuación.



**Figura 22:** primer montaje para ASA (a)



**Figura 23:** Primer montaje para ASA (b)

Se puede ver como se ha incorporó el microcontrolador LPC2148 al vehículo. Todas las placas que contiene se diseñaron y se fabricaron manualmente.

Para la dirección, se decidió usar un servomotor modelo Hitec-HS311, un servo de tamaño y funciones estándar, este sería el mismo que se usase para la construcción del robot.

La alimentación es un simple paquete de pilas para el microcontrolador y la batería antigua que traía el coche. Totalmente todo desestimado a posteriori para centralizar todo en una misma batería. Este prototipo de desmontó y se procedió a construir en sí mas íntegramente el vehículo.

Lo primero que se hizo fue reforzar el chasis, con lo que se creaba un buen soporte para todo el peso que iba a tener que soportar. Ésta se creó alineando bien la base, debido a que traía una articulación en el medio. Se le colocaron dos planchas de madera arriba y abajo a modo de bacadillo y atornillando una con la otra dejando el chasis en medio. Aparte de la robustez que da, al usar madera se está consiguiendo aligerar peso del vehículo. También estaba la posibilidad de usar aluminio o metacrilato. La madera respecto a estos dos proporcionaba ligereza y robustez.



Al realizar el chasis del coche había que colocarle el servomotor, así que había que crear el hueco para éste, debido a que estaba a mitad de donde tenía que colocarse la plancha de madera. Se midió y se preparó dicho espacio. La posición en la que se quería colocar era la misma en la que ya estuviese el servo que traía el coche teledirigido de fábrica.

Al servo que se quería colocar, hubo que hacerle una pequeña modificación, y fue cortarle las pestañas de sujeción a los tornillos, dado que estorbaban y con ellas no se podía colocar en su sitio correspondiente. Esta se aprecia en la figura



Figura 24: modificación realizada al servomotor

Aunque diese la impresión de que no iba a tener un ajuste fiable, se soportó por presión y quedó bastante rígido, quedando finalmente la base como se aprecia en las siguientes imágenes.

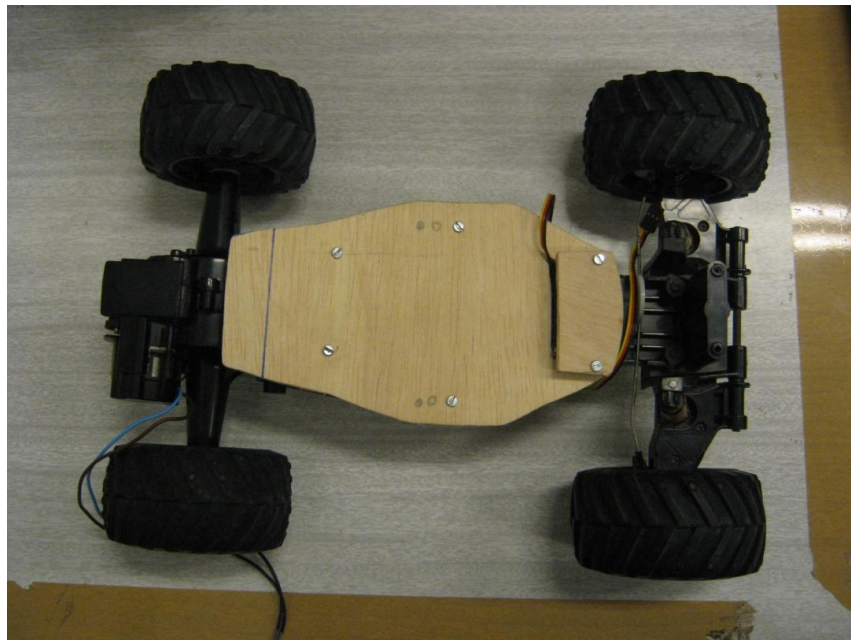
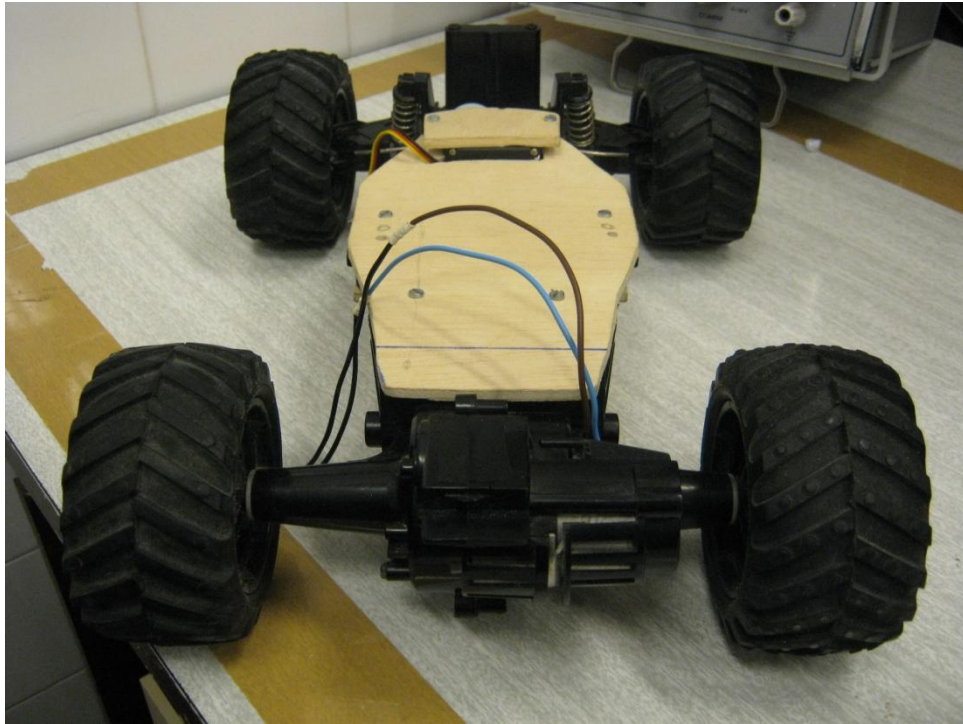


Figura 25: base chasis (a)



**Figura 26:** base chasis (b)



**Figura 27:** base chasis (c)

Podemos ver como el servomotor ha quedado en ese hueco especialmente diseñado para él. Este es el resultado final de la base sobre la que se construiría el vehículo completo.

El haber utilizado madera, proporciona una buena base sobre la que trabajar para atornillar nuevos elementos a ésta, si fuese aluminio sería una tarea más dura.

Para atornillar las dos planchas de madera se han utilizado tornillos de 3mm extra largos del laboratorio y apretándolos con tuercas, una vez hecho se comprobó la rigidez que tenía el resultado.

Apreciamos en las imágenes como el servo queda resguardado y bien asegurado en ese hueco. Con su pieza de madera atornillada arriba suya haciendo presión para que no se mueva y pueda proporcionar un manejo de las ruedas fiable.

En la parte opuesta al servo vemos dos cables, uno marrón y otro azul, que son los cables de alimentación del motor, estos son los que más tarde irán al driver del motor. Disponían de una longitud suficiente para trabajar con comodidad.

Antes de seguir con el montaje se comprobó el funcionamiento del motor probándolo con una fuente de alimentación externa.

Si observamos delante del servomotor, se ven dos muelles que es la amortiguación delantera que traía el vehículo original, entre ellas, se observa un soporte el cual llevaría sujeta en él la carcasa original, este será utilizado para el siguiente paso, la plataforma del primer piso.

### **1.8.2.2. Plataformas superiores**

Una vez que se tenía la base, era el momento de situar toda la electrónica en el vehículo. Para ello se diseñó una plataforma que tuviese el suficiente espacio para albergar todas las placas, conectores y demás elementos que fuese a llevar, sin olvidar que es sobre esta plataforma sobre la que se apoyará una segunda, en la que se situaría lo que denominamos cabeza articulada.

La herramienta utilizada para el diseño de esta primera plataforma fue VCarve. Este software permite el diseño en el ordenador de una pieza cuya fabricación se va a llevar a cabo en una máquina de control numérico.

No se va a explicar el funcionamiento del software dado que queda fuera del objetivo de este proyecto. Antes de diseñar en ordenador, se diseñó en papel y luego en cartón, para tener una primera idea de cómo quedaría la placa que se quisiese hacer. También esto permite hacer modificaciones rápidas que si se quisiesen hacer en una pieza ya mecanizada implicaría la fabricación de nuevo dicha pieza.



La electrónica que debía de llevar encima debía de ser la elegida para el vehículo. Se ha explicado con anterioridad cada una pero se vuelven a mostrar.

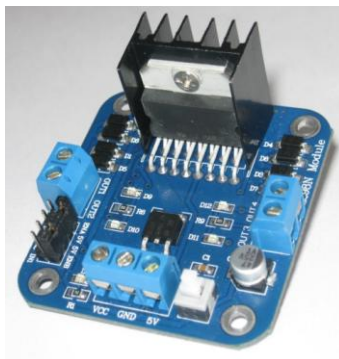
- BeagleBoard xM



- LPC2148

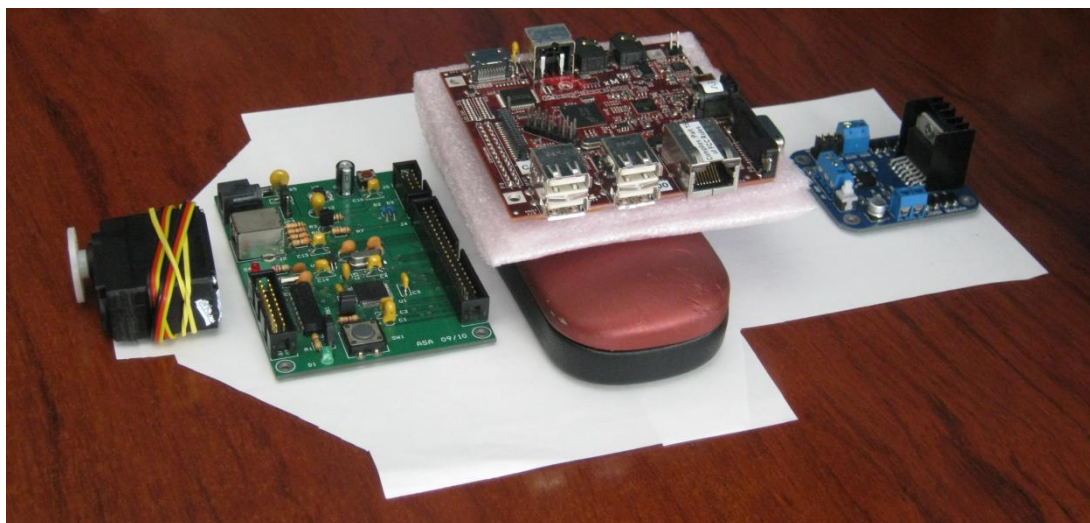


- Módulo L298



La forma de colocar las placas en la pieza que se hiciese debía de ser tal que permitiese conexiones cómodas, sin que estorben unas con otras y pudiendo atornillarlas bien.

Una primera forma que se pensó para colocar estas placas en el chasis del coche, fue como se muestra a continuación:



**Figura 28:** primera idea de montaje

La funda morada, hace las veces de batería. Apreciamos que en este montaje todo quedaría bastante apretado. La batería, es un elemento que debe de ser de fácil acceso, puesto que habría que cargarla numerosas veces, así que esta no debería de ser difícil su puesta así como el quitarla.

A su vez, que la superficie, quedaba a un nivel excesivamente bajo, más que la altura de las ruedas, con lo que se limitaba mucho el espacio en la superficie de la plataforma para futuras implementaciones que se llevasen a cabo.

Una idea que sí se sacó en claro, era que el disipador que tiene incorporado el L298, es el elemento que más calor iba a disipar de todo el vehículo, y como el movimiento iba a ser en avance generalmente, el módulo del L298, debía de estar en una posición retrasada, respecto a lo que fuese la parte delantera, para que el calor que se fuese generando, saliese hacia atrás y no afectase al coche. El hecho de ponerlo en una posición retrasada, también beneficiada por que el motor del coche también estaba situado de forma similar, con lo que se acortaba la longitud de los cables.

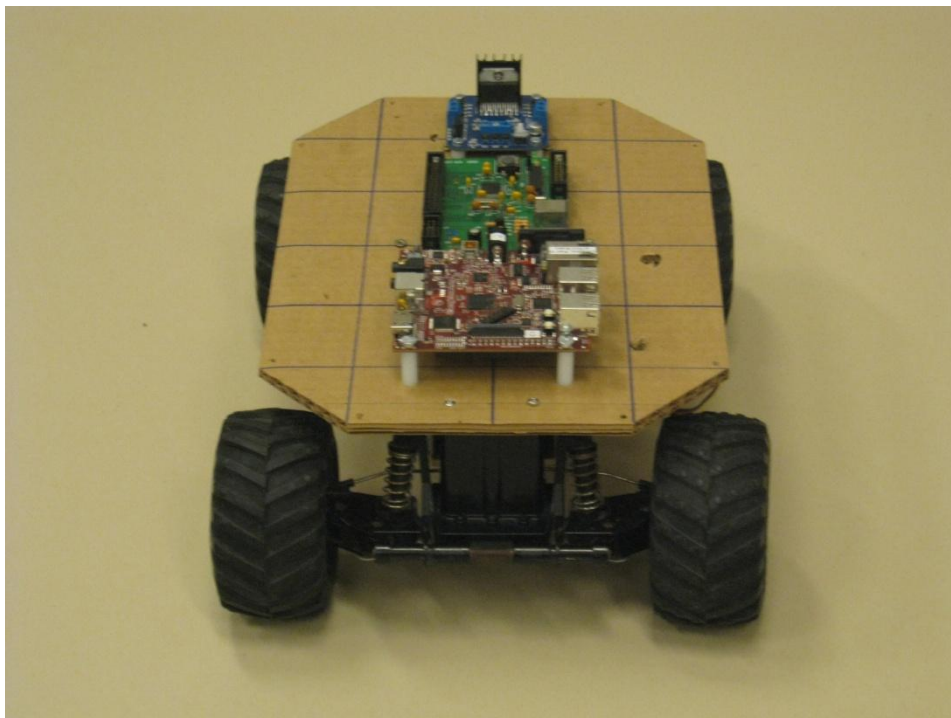
A partir de este montaje, también se observó que la batería no era conveniente que estuviese cerca de la electrónica, ya que la batería que se iba a usar, una batería de Li-Po, es relativamente peligrosa, hasta el punto de llegar a explotar. Así que se pretendió que hubiese una distancia o algún tipo de barrera entre la electrónica y la batería, a ser posible, que la batería tuviese su sitio para colocarse resguardado. Teniendo en cuenta que debe de ser de fácil acceso para ponerla y quitarla.

Con estas ideas en claro, se debatió cómo diseñar la plataforma. Se optó por construirla en una posición más elevada, con lo que se ganaría espacio, estando en una posición superior respecto a las ruedas. La idea fue una superficie octogonal. Lo bastante grande para albergar a las placas y demás elementos que se quisiesen montar encima.

El primer prototipo que se diseñó siguiendo estas pautas fue construido con cartón y probado para ver su acabado y como encajaba en el chasis.

Para instalarlo, se utilizaron separadores hexagonales de 50 mm, además de usar soportes que trajese el chasis para sujetar la anterior carcasa.

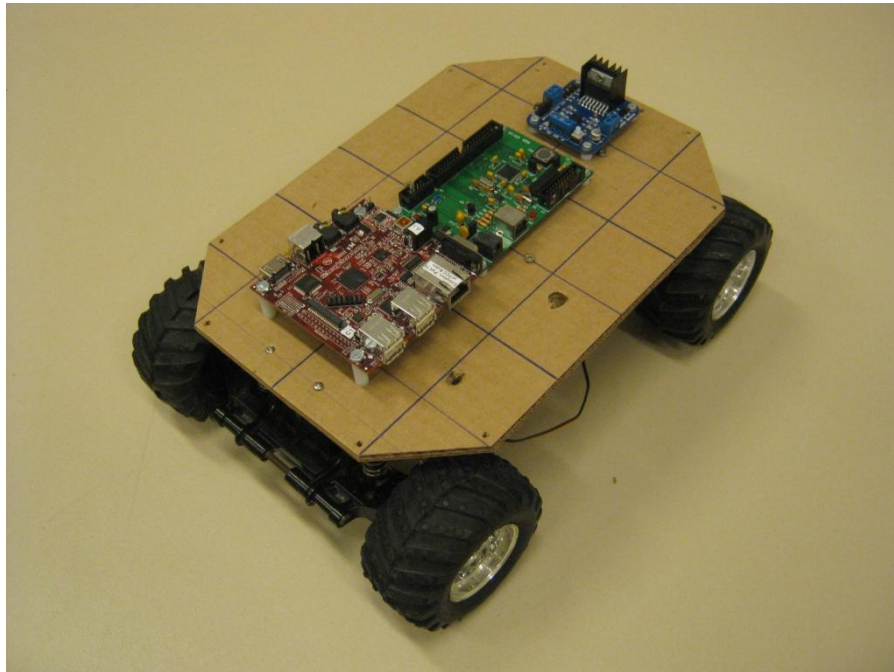
Una vez hecho el prototipo se colocó inmediatamente la electrónica sobre esta para ver si se había llevado a cabo un buen diseño como se muestra en las imágenes. De no encajar, se diseñaría de nuevo supliendo los errores que pudiese haber.



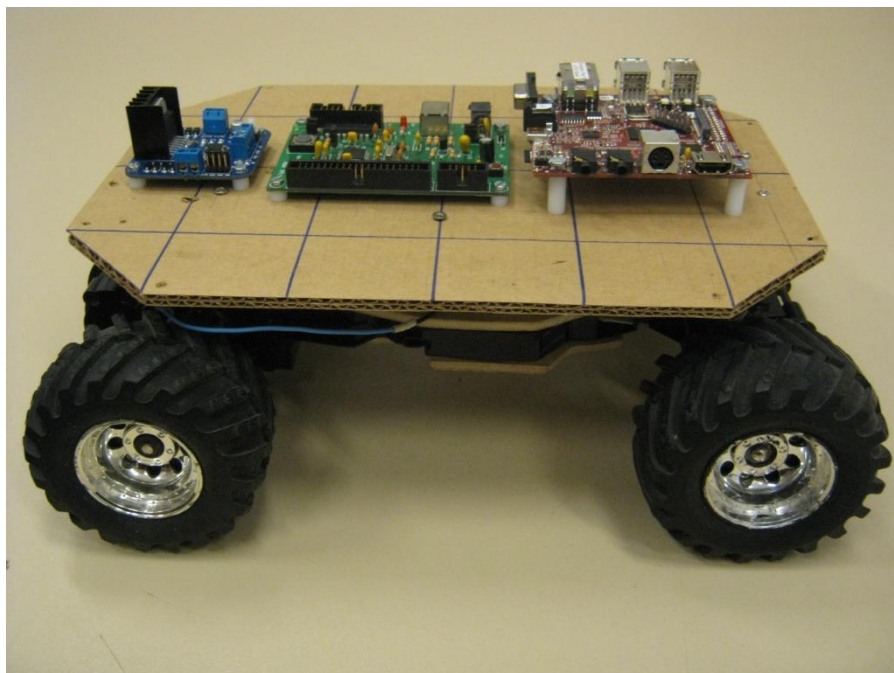
**Figura 29:** Primer prototipo plataforma principal, cartón. (a)

La disposición decidida para colocar las placas fue como se muestra en la figura 1.27, alineadas una detrás de otra. Se colocó la BeagleBoard en primer lugar, tras ella el microcontrolador y por último el módulo del L298. El hecho de poner la Beagleboard-xM delante viene a que el conector de pines de entrada y salida, está situado en un lateral, al poner el micro tras ella, este conector quedaría cerca y se podría llevar a cabo una cómoda conexión sin tener que cablear demasiado. Que el microcontrolador estuviese en el centro permite estar cerca de todos los puntos del vehículo. Se sigue la pauta de colocar atrás el L298, para que así todo el calor pueda salir desprendido bien hacia detrás.

Como vemos, la forma octogonal permite un diseño moderno, a la vez que proporcionando suficiente espacio para albergar sucesivos elementos y en un nivel superior a las ruedas, así que el giro de estas no implicaría peligro alguno para las placas.



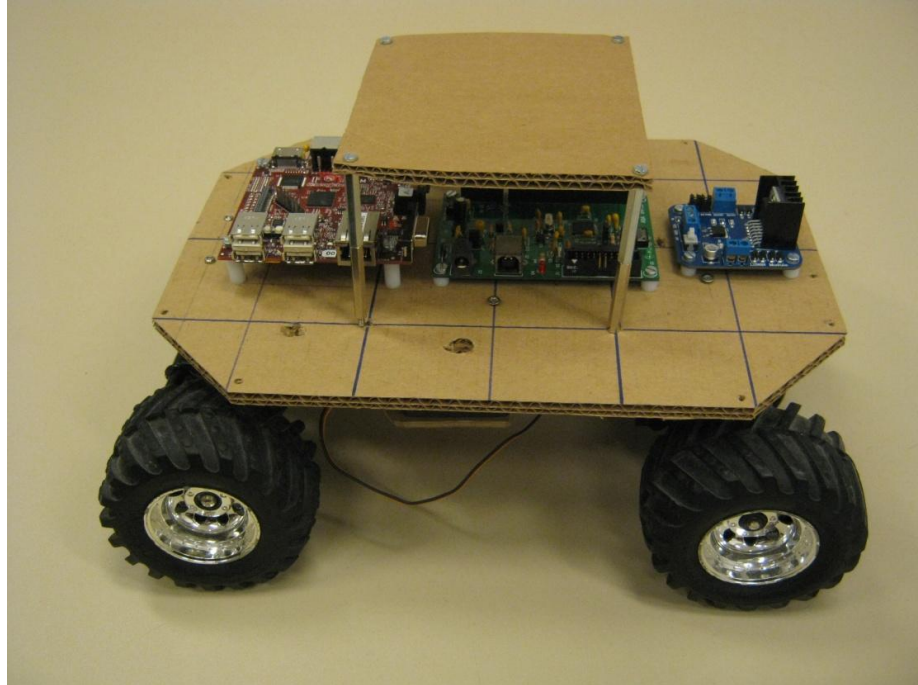
**Figura 30:** Primer prototipo plataforma principal, cartón (b)



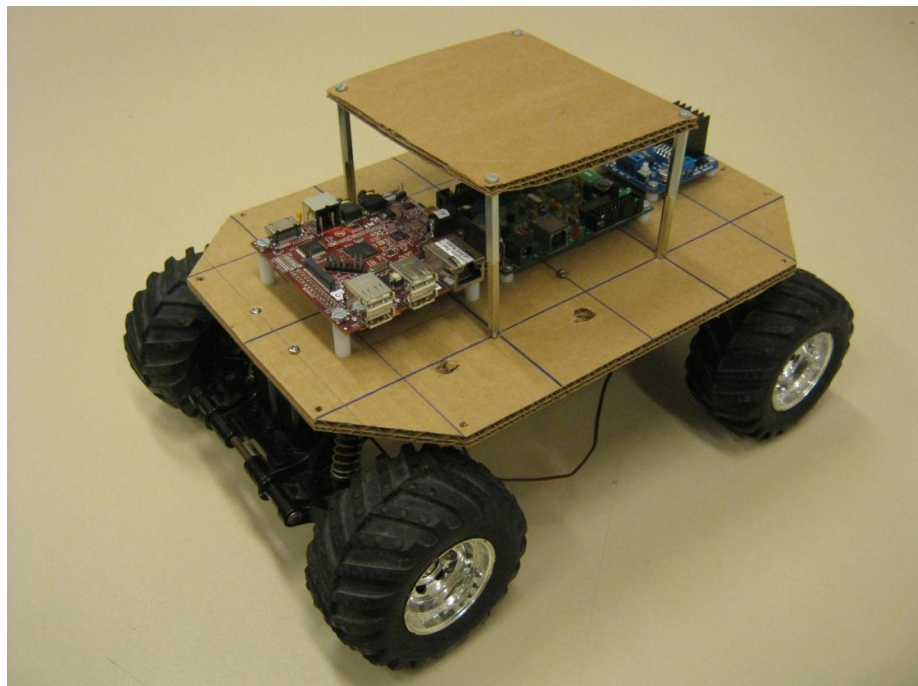
**Figura 31:** Primer prototipo plataforma principal, cartón



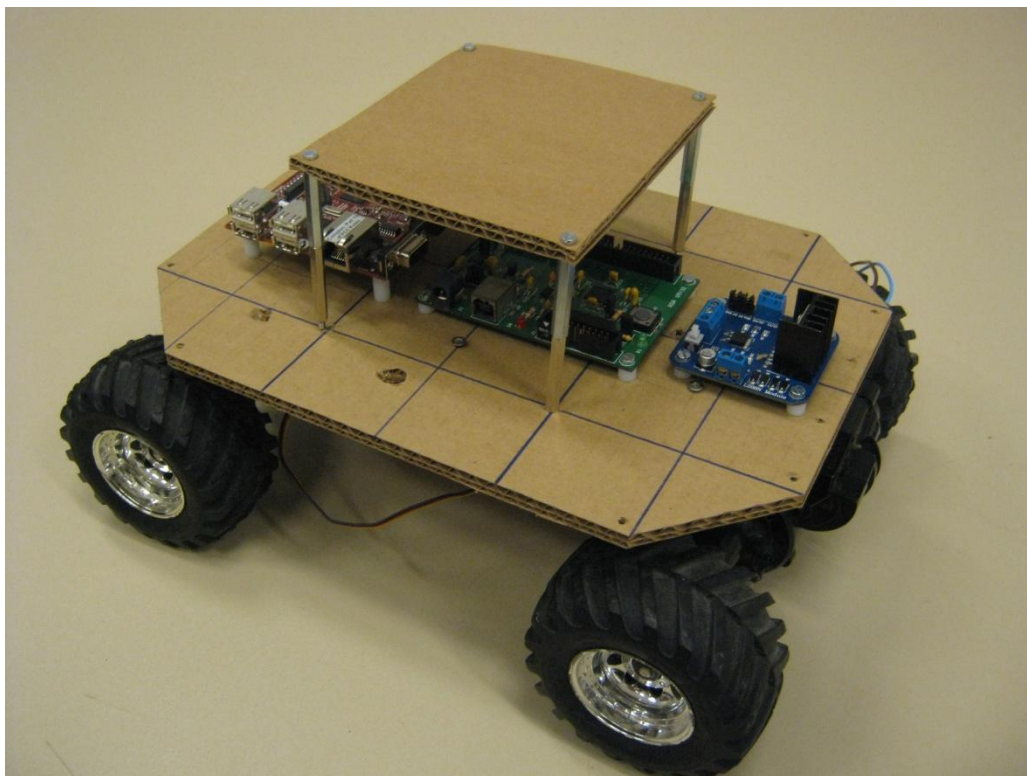
Al hacer este modelo, se pensó como implementar todos los sensores y la instalación de un piso superior. Al seguir probando sobre cartón, se improvisó lo que sería el piso superior, se fabricó de manera rápida, en él se instalaría la articulación. El resultado fue el siguiente:



**Figura 32:** Plataformas con pisos 1° y 2°. (a)



**Figura 33:** Plataformas con pisos 1° y 2°. (b)



**Figura 34:** Plataformas con pisos 1º y 2º (c)

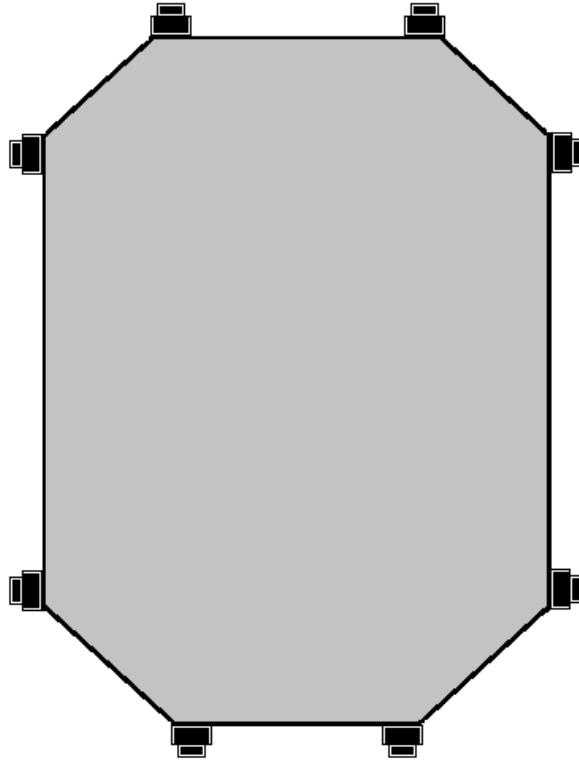
Este piso superior debía de soportar toda la articulación del vehículo, es decir, dos servomotores junto con el peso propio de la cámara, además de toda la tornillería para sujetarla.

Debía de estar diseñada de forma, que no molestase nada al realizar un movimiento. También debía de estar bien situado para que nada obstaculizase con la visión de la cámara. Si esto llegase a suceder, no se estaría aprovechando al máximo las posibilidades de la cámara, provocando problemas al procesamiento de imágenes ya que esta zona debía de obviarse al realizarlo.

A continuación se pensó en cómo colocar los sensores de distancia. Para un mejor tratamiento del entorno, había que tener cubiertos los cuatro lados del vehículo, delantero, trasero, derecho e izquierdo. La elección fue colocar los sensores GP2D12 por la periferia del vehículo, estos sensores miden en un punto, no como lo hace el radar, que ve un cono completo, con lo que son mucho más fiables para el propósito que se quería lograr.

Su colocación tenía que permitir el comprobar el estado de una pared, si es posible seguirla y detectar obstáculos en todas direcciones así como su orientación respecto a ellos.

Esto se logró colocando los sensores de la siguiente forma:



**Figura 35:** Colocación de sensores infrarrojos

Se pensó también en taladros para cables y conectores.

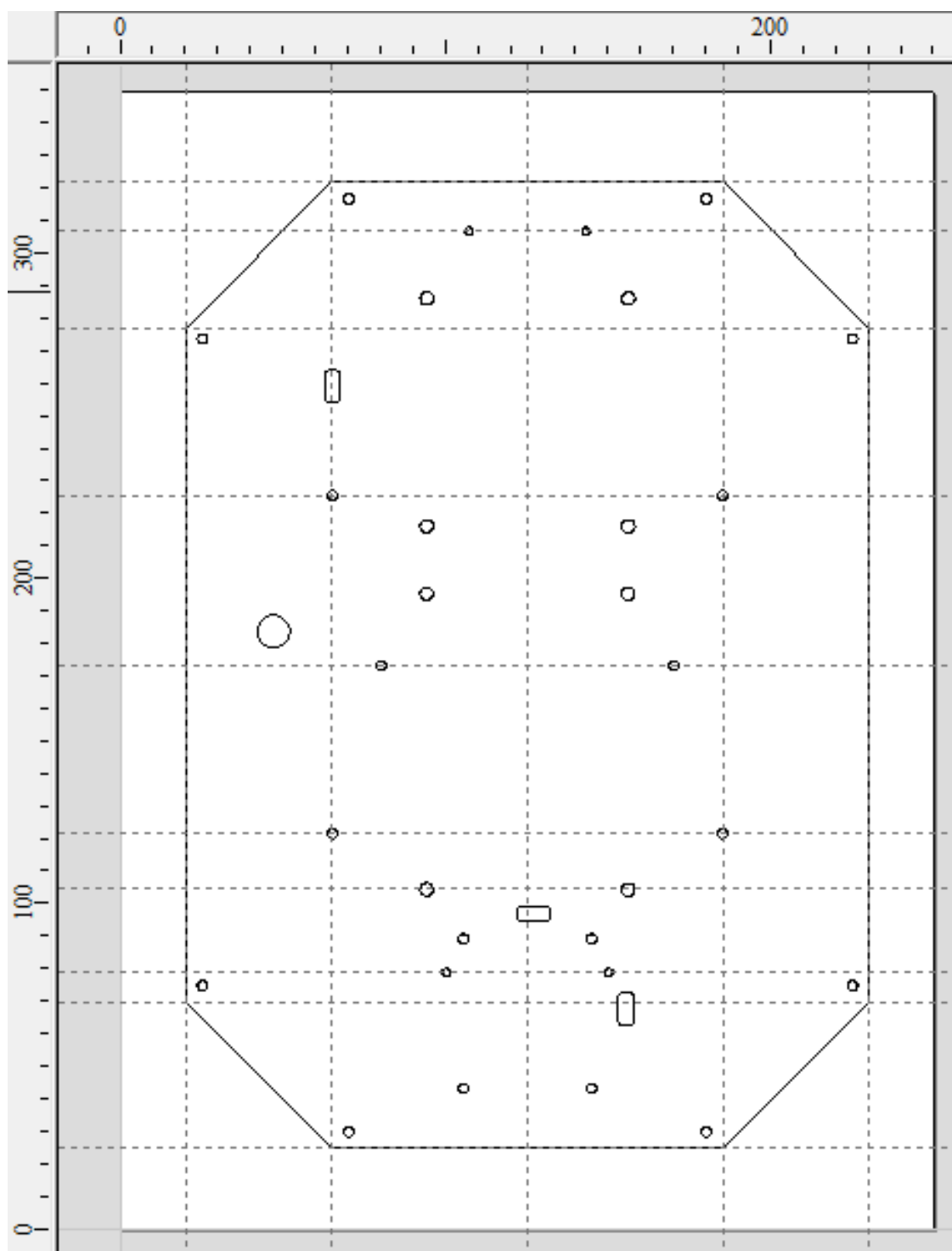
Una vez que se probó y todo encajaba bien, se llevó a cabo su fabricación en metacrilato.

Para desarrollar esta pieza, la fabricación se ha llevado a cabo en la máquina de control numérico del laboratorio de robótica en una plancha de metacrilato.

Hubo que aprender anteriormente a usarla, pero unas simples sesiones de aprendizaje bastan para aprender su funcionamiento.

Una vez mecanizada se hizo el mismo proceso que cuando se fabricó en cartón, instalar la electrónica y probar que todo encajaba bien, aunque si había un fallo se modificaría sobre ella misma, dado al tamaño que tenía, no se iba a fabricar otra.

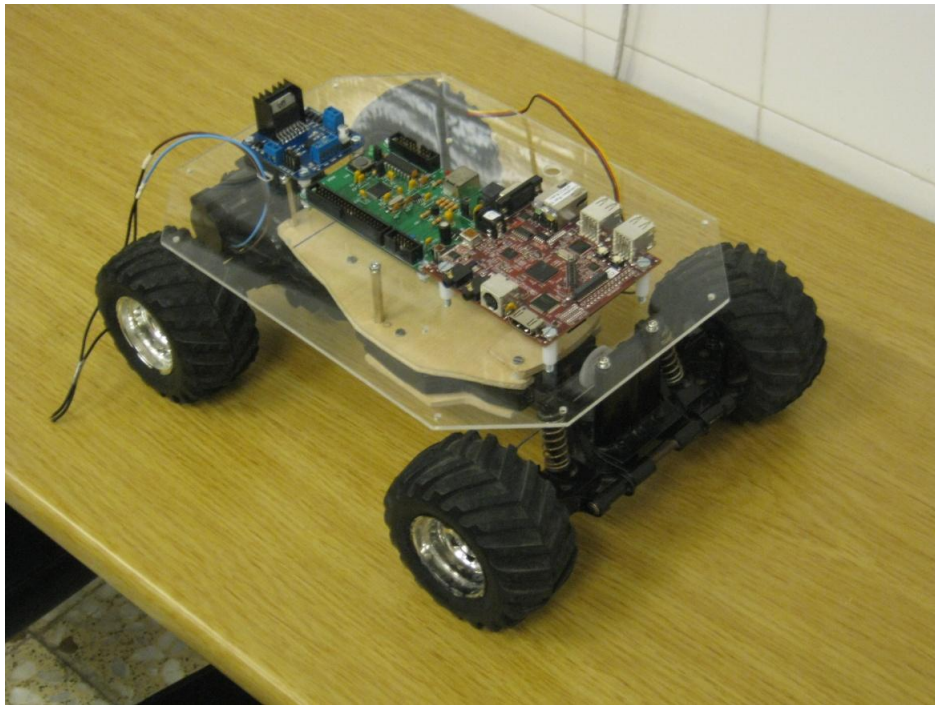
La pieza diseñada primero con VCarve fue la siguiente:



**Figura 36:** Diseño plataforma principal



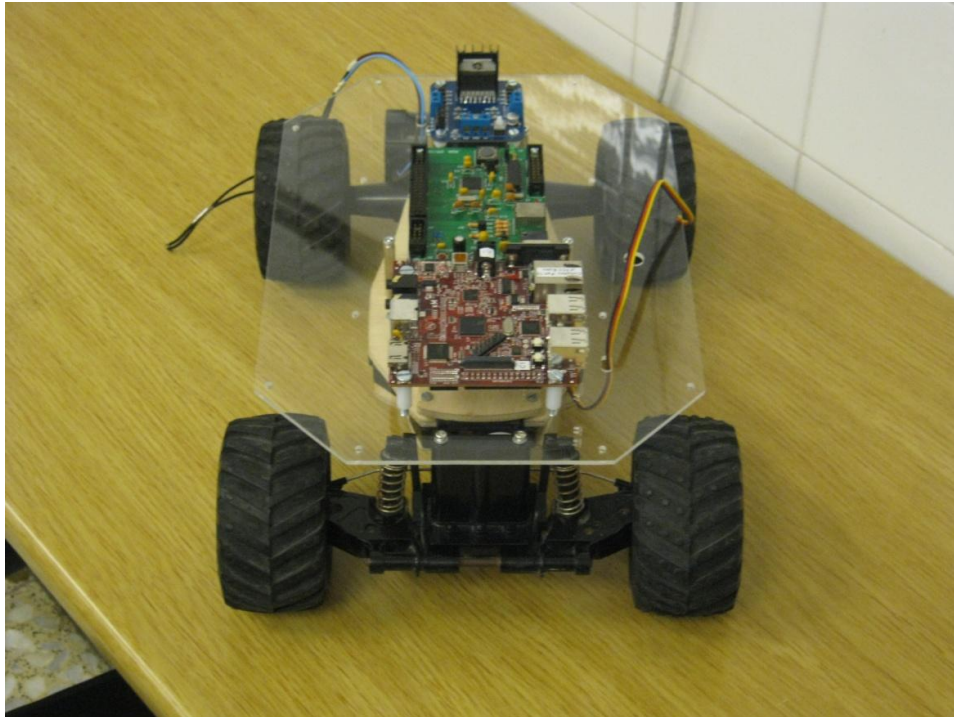
Cuando se fabricó y se montó la electrónica tal y como se hiciese en el cartón, el aspecto que quedó fue el siguiente:



**Figura 37:** Base principal de metacrilato (a)



**Figura 38:** Base principal metacrilato (b)



**Figura 39:** Base principal de metacrilato

Una vez que se montó, se vio que la electrónica encajaba bastante bien, con lo que se fabricó la parte superior. Que encajó a la perfección, con lo que se fue montando todo el resto del vehículo.

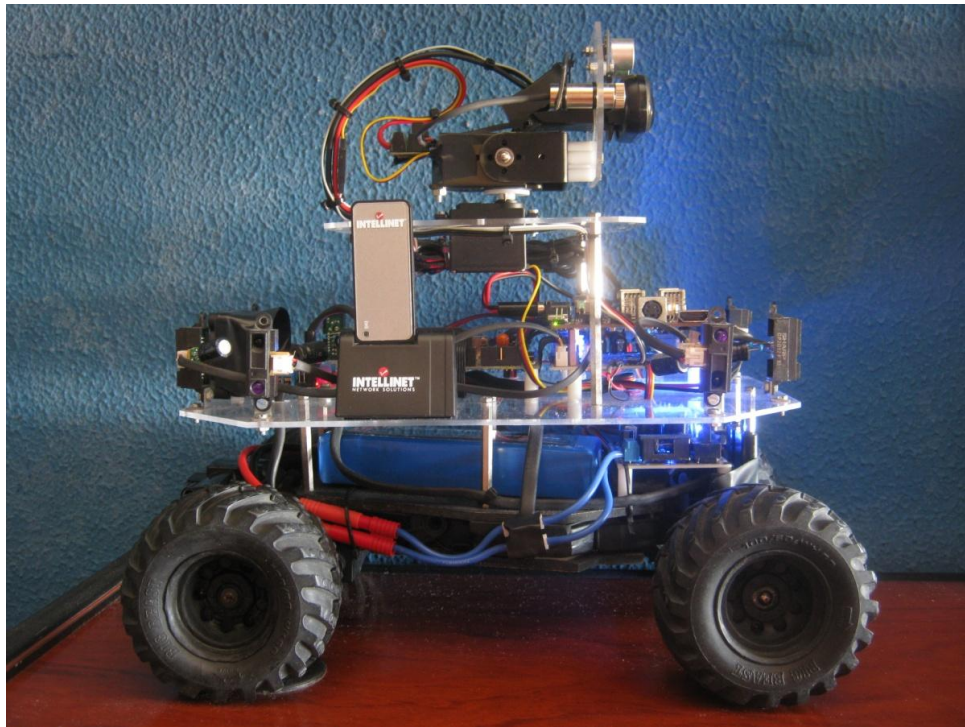
Debido a la falta de cámara, no se pudieron ir tomando suficientes fotos para apreciarlo, así que se mostrarán imágenes globales especificando lo que se esté describiendo.

A parte del trabajo requerido en trabajo del coche, muchas veces en las que se estaba en posesión de cámara se olvidó el documentar por querer avanzar en el proyecto.

Lo próximo que se hizo fue terminar de montar el resto del coche, sensores, cámara láseres, Wi-Fi, alimentación, batería etc, etc...



### 1.8.2.3. Montaje final

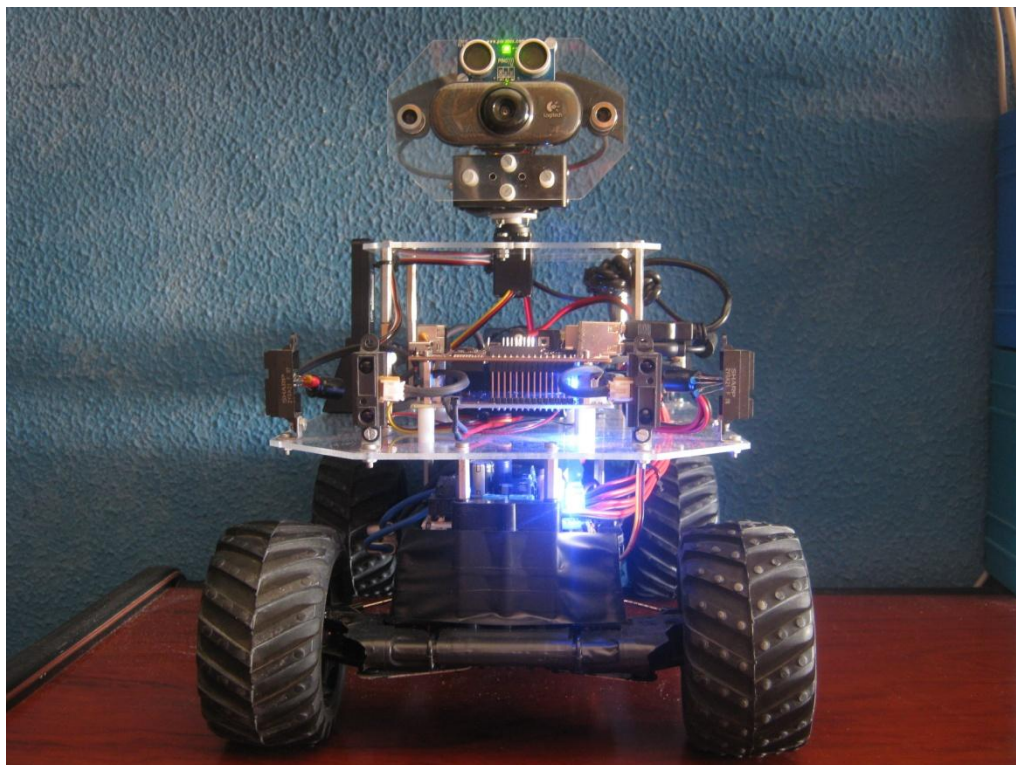


**Figura 40:** vista lateral derecho

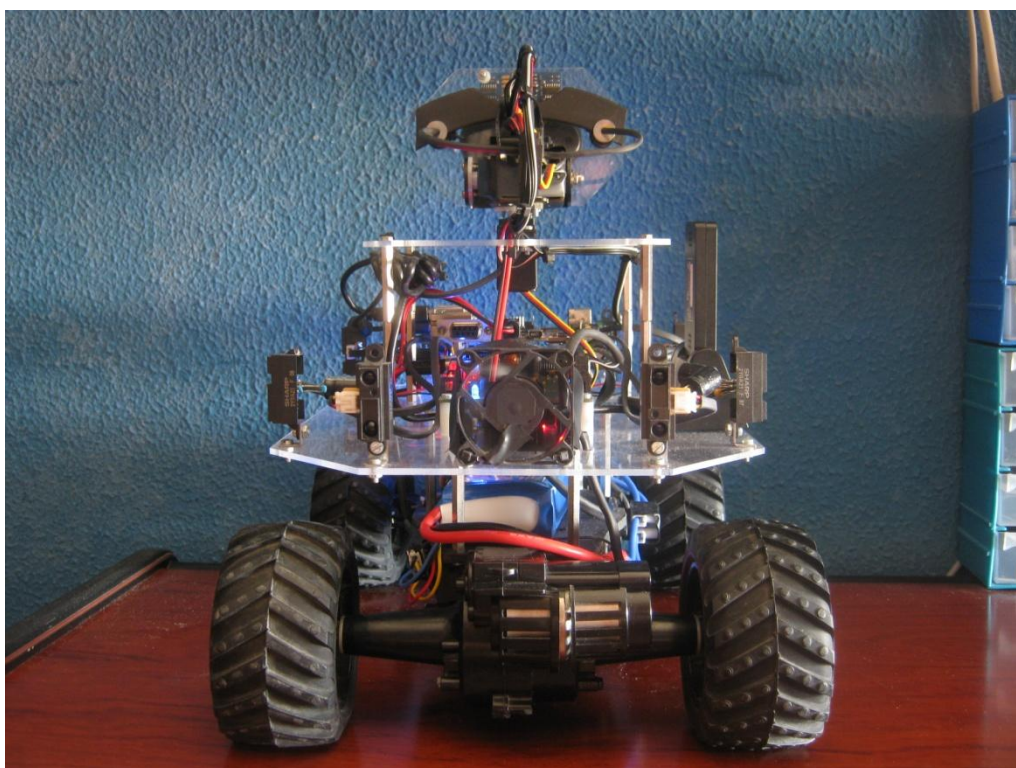


**Figura 41:** vista lateral derecho

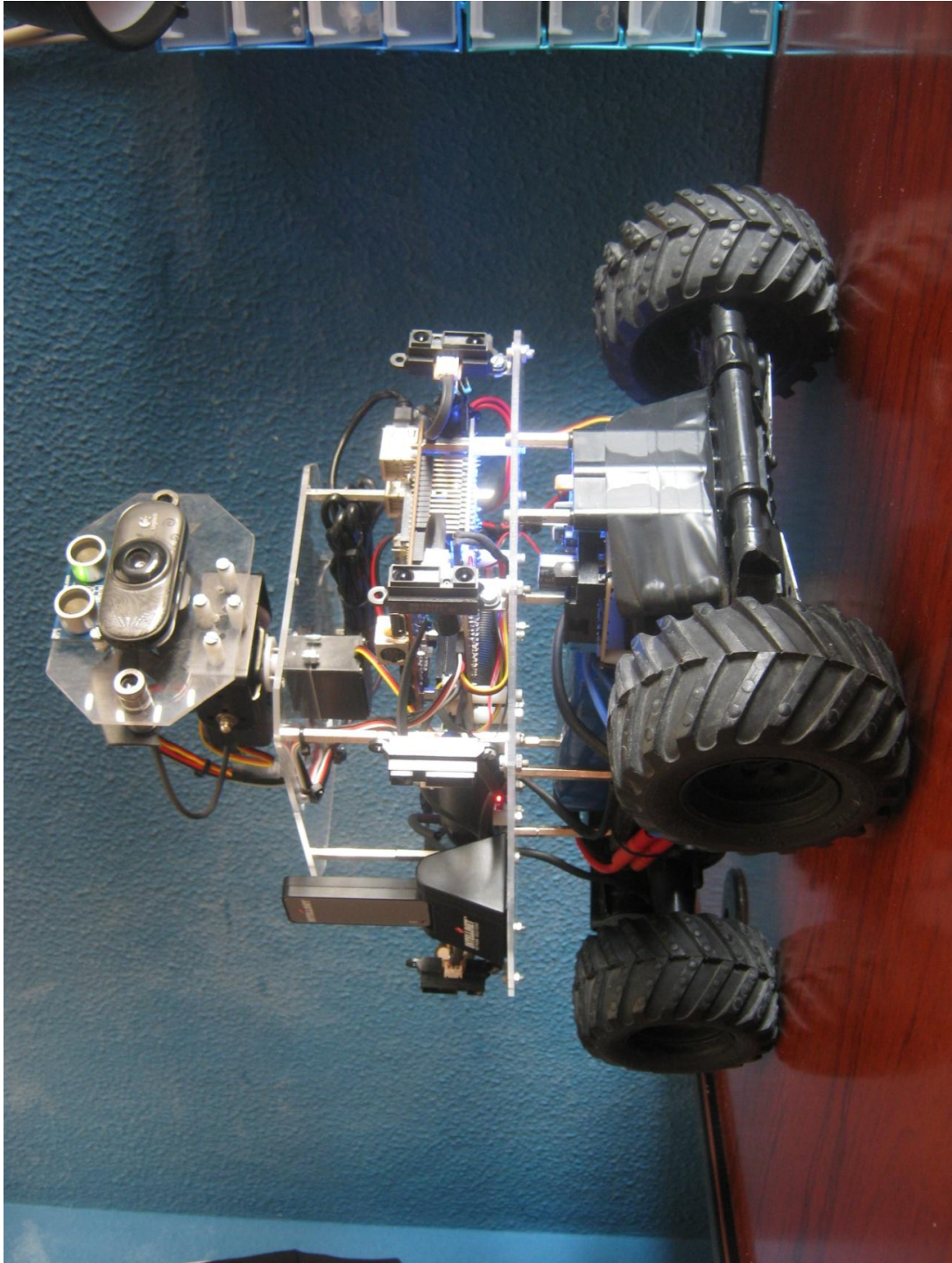




**Figura 42:** vista frontal



**Figura 43:** vista trasera



**Figura 44:** visión general



#### 1.8.2.4. Cabeza articulada

Se montó la articulación en el segundo piso de metacrilato, quedando a suficiente altura para que el servo de movimiento horizontal no tocara con la electrónica que está justo debajo suya.

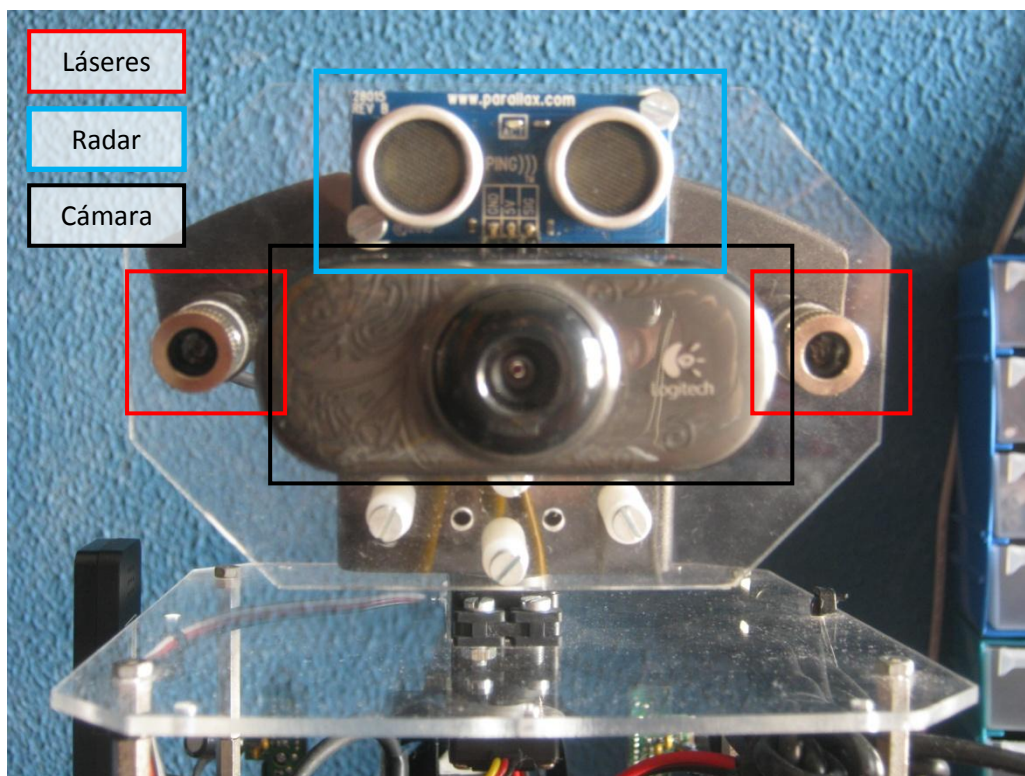
Se le montaron los láseres en paralelo, uno a cada lado de la cámara y se conectaron por detrás utilizando una clema.

Se instaló la cámara fácilmente, pasando el cable sin problema por detrás.

El sensor por ultrasonidos se instaló justo encima de la cámara, un poco forzado el cable pero quedó bastante bien, así a donde mirase la cámara se podía medir la distancia.

Todos los cables provenientes de la cabeza se pasan por un taladro bastante amplio situado en la parte trasera.

A estos cables se les dejó bastante longitud para que el movimiento de la cabeza no les afectase y tuviese gran libertad para moverse.



**Figura 45:** cabeza articulada frontal

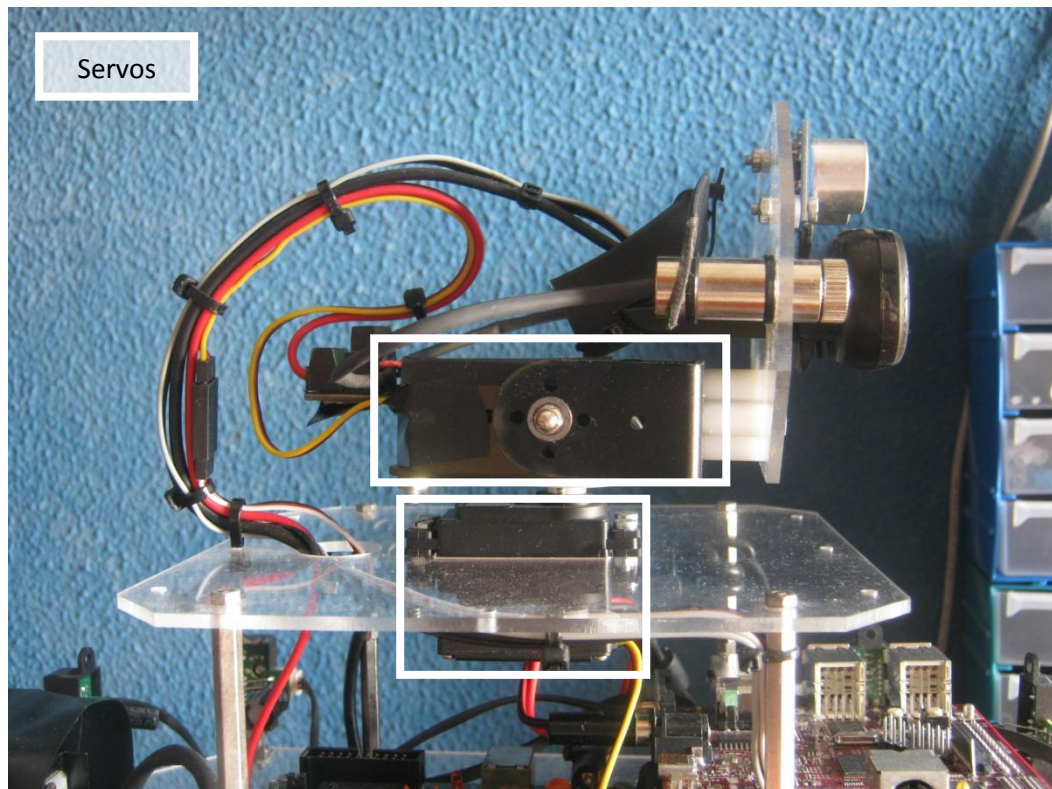


Figura 46: cabeza articulada lateral

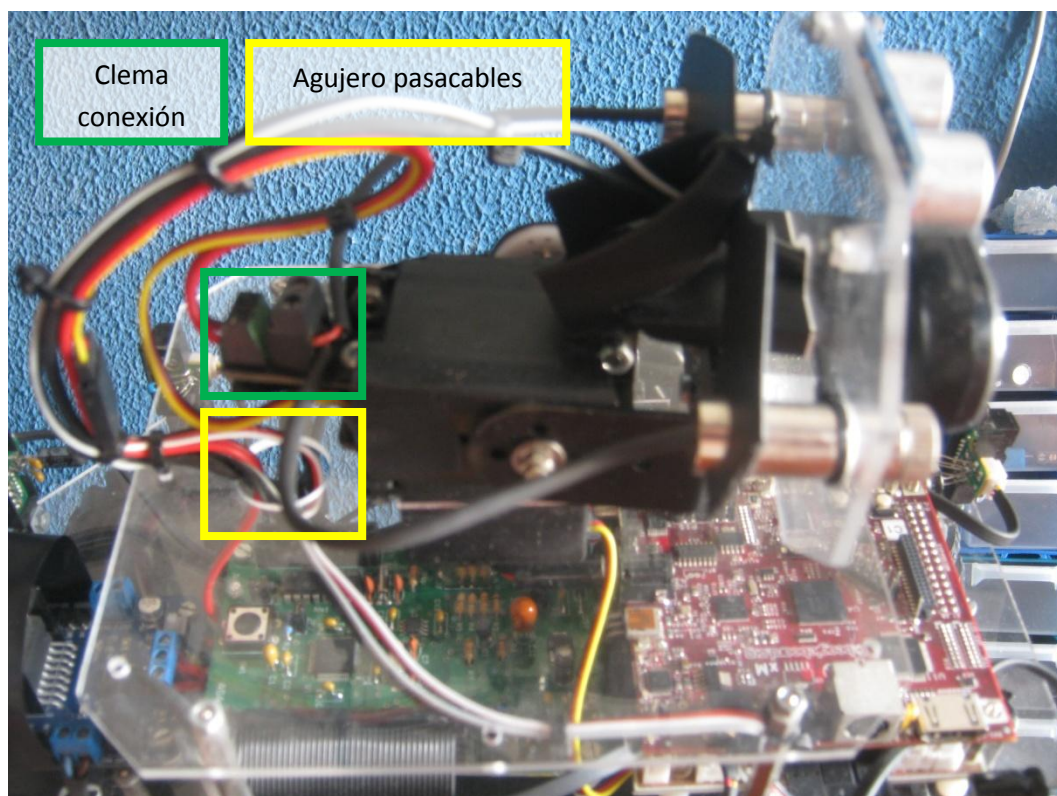
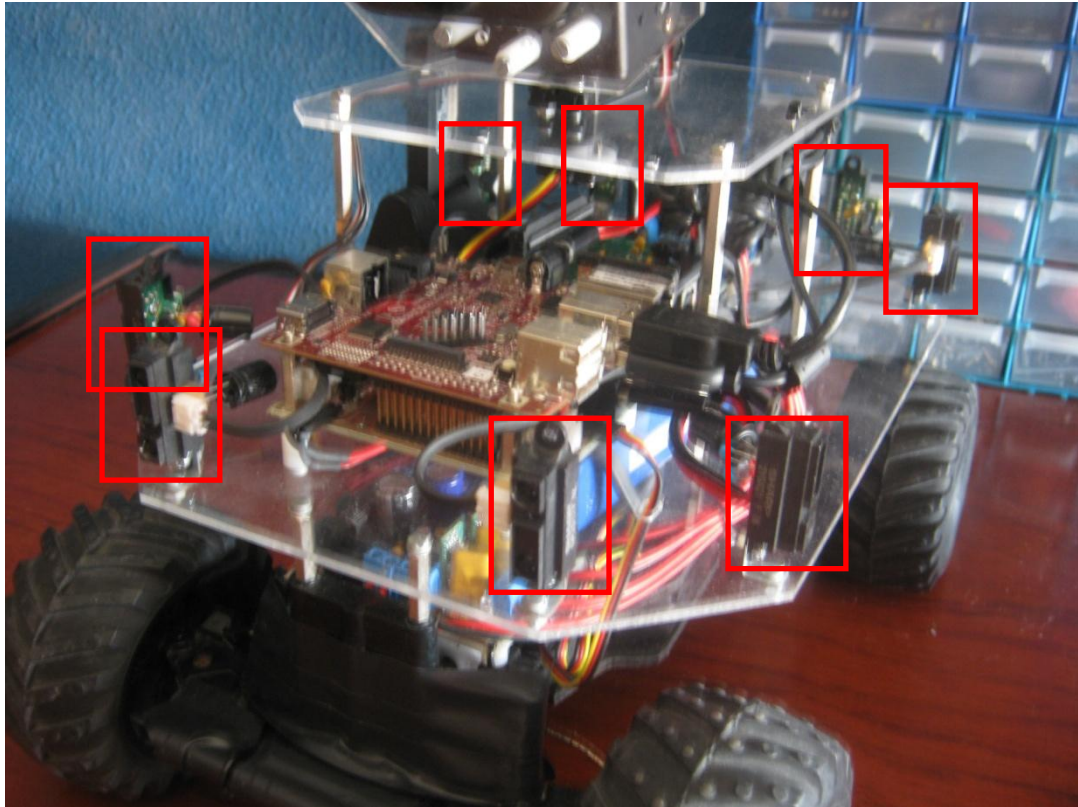


Figura 47: cabeza articulada superior



### 1.8.2.5. Sensores infrarrojos

Los sensores se dispusieron por parejas en cada uno de los lados del coche, a ser, delante, detrás, derecha e izquierda. Así se tendrían al menos dos por flanco.



**Figura 48:** Disposición de los sensores infrarrojos



### 1.8.2.6. Ventilador

Debido al gran calor disipado por el integrado L298 ha sido conveniente incorporarle un pequeño ventilador disipador de calor para poder evacuarlo así con más facilidad.

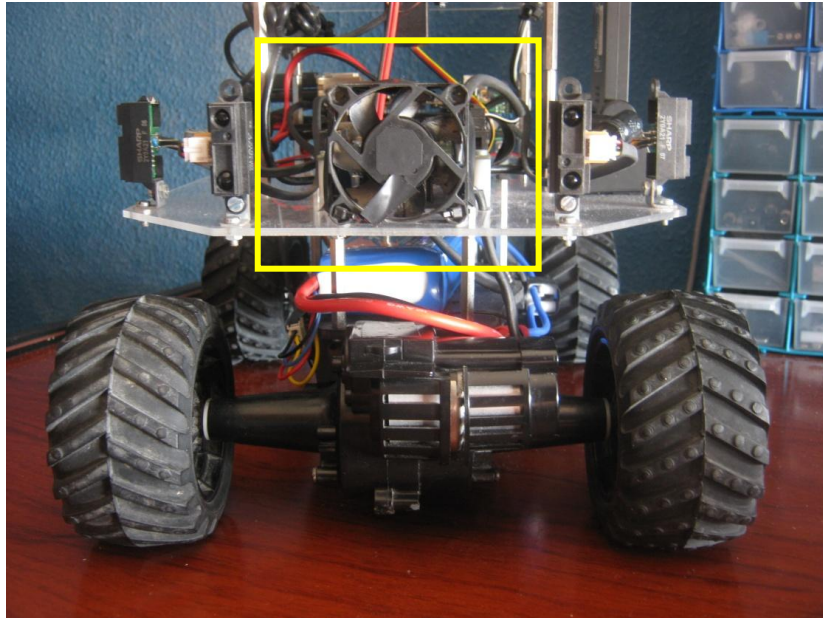


Figura 49: posición del ventilador

### 1.8.2.7. Módulo Wi-Fi

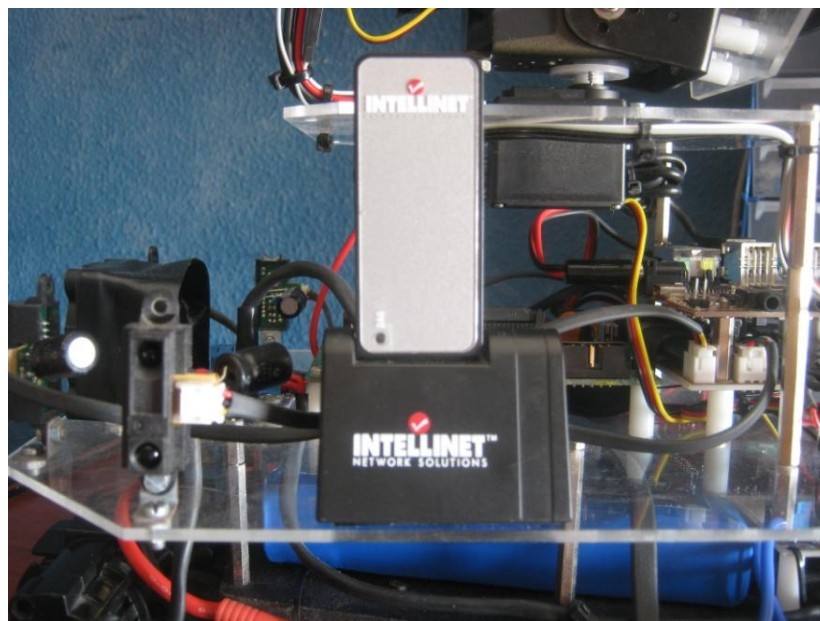


Figura 50: Colocación del módulo Wi-Fi

### 1.8.2.8. Batería y fuente de alimentación

La batería se dispuso en el hueco pensado para ello y la fuente de alimentación se colocó en la parte delantera pero debajo del soporte principal, es de aquí de donde partirían todos los cables de alimentación.

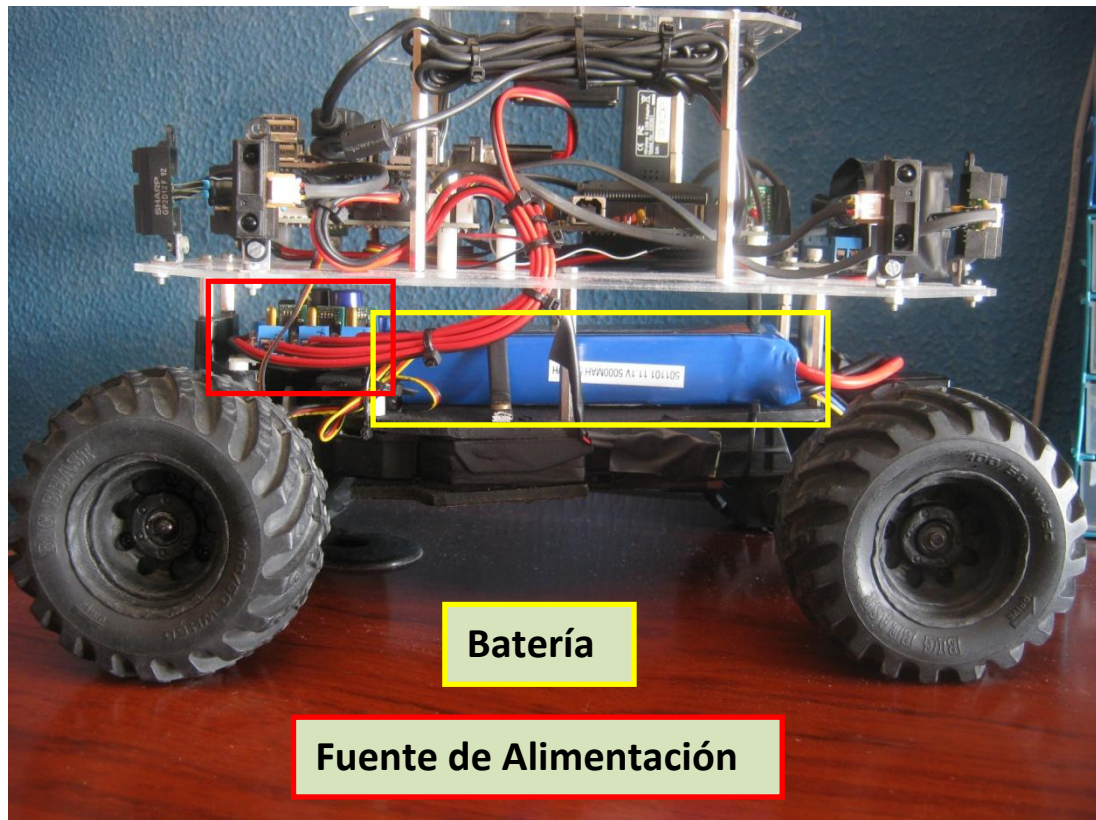


Figura 51: colocación de la batería

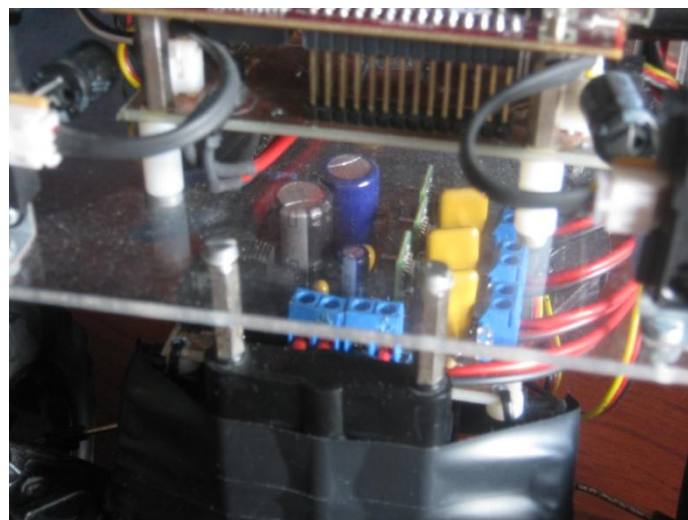


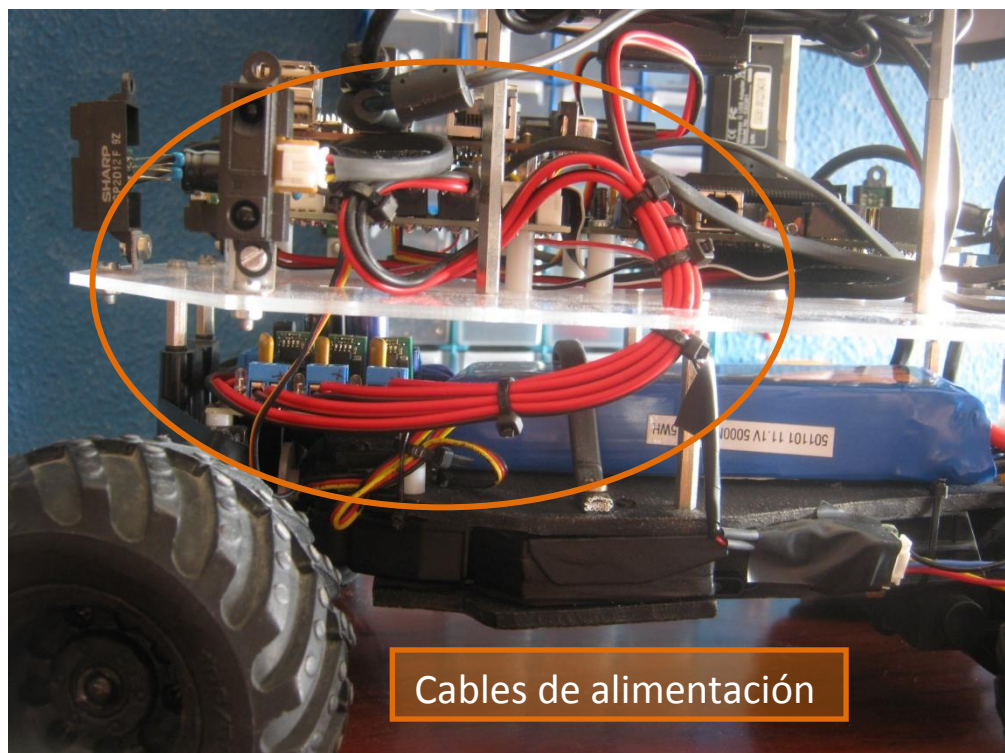
Figura 52: vista frontal de la fuente de alimentación





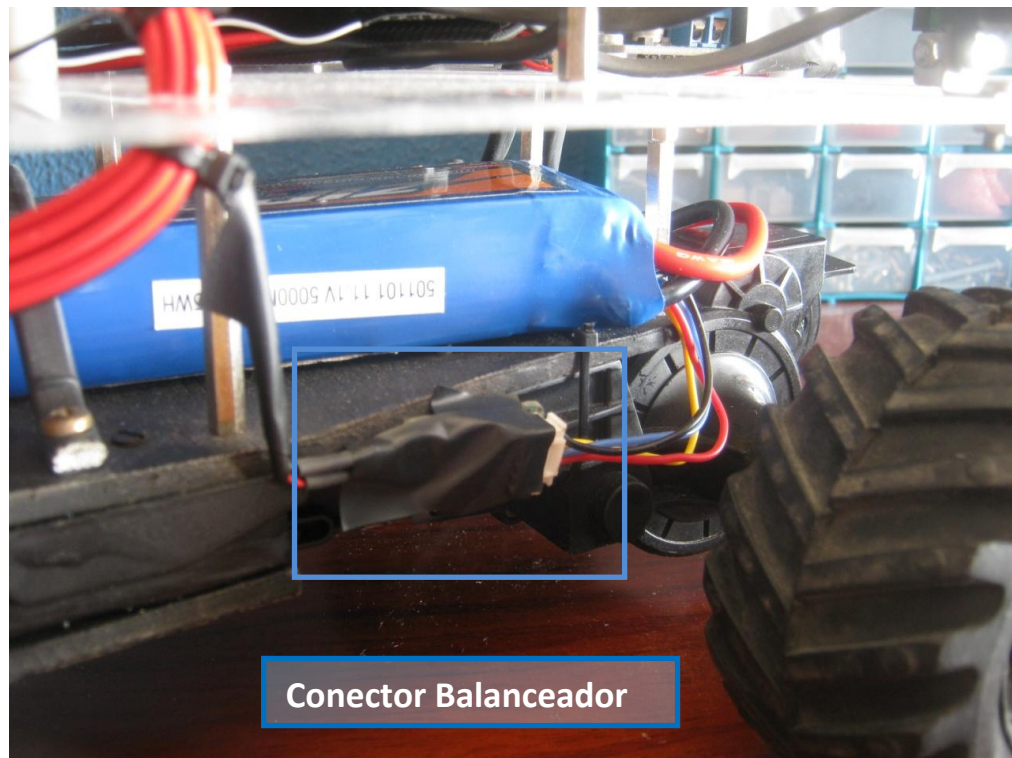
Conector de Batería

**Figura 53:** conector para la batería



Cables de alimentación

**Figura 54:** cables de alimentación



**Figura 55:** conector para balanceador de la batería

### 1.8.3. Descripción del sistema

El sistema completo debe tener una estructura que permita ser gobernado por un usuario, pero que permita que el propio vehículo pueda tomar decisiones en algún momento dado. Ha de esclarecerse bien cada parte de la que consta, describiendo bien la misión que tiene cada una, los elementos de que se compone, como funciona y como se relaciona cada una de estas partes con las demás. Hay que definir qué tipos de relaciones se establecen, todo esto está explicado a continuación, en donde se detalla cómo se ha configurado el sistema.

#### 1.8.3.1. Definición general del sistema

Se trata de un sistema jerárquico, el cual se divide en tres grandes bloques.

**Grupo 1:** Puesto de observación y control del vehículo. Desde él se le dan las órdenes al vehículo a ejecutar y lo que hacer en cada momento. Este grupo gobierna a todo el sistema por completo.

**Grupo 2:** El cerebro. Recibe las órdenes a seguir del grupo uno y se encarga de la toma de decisiones cuando se esté trabajando de forma autónoma. Recibe imágenes de la cámara y a su vez gobierna al grupo tres.

**Grupo 3:** El cuerpo. Hace lo que le mande el cerebro. Su trabajo es leer los sensores y mover todos los servomotores y el motor en base a lo que mande el grupo dos.

Al tratarse de un sistema jerárquico, el que está en un mayor nivel es el que más poder tiene. Cumpliéndose las dos siguientes leyes:

- Un grupo inferior, no puede mandar sobre un grupo superior.
- Un grupo superior, no puede mandar sobre un grupo inferior, si entre estos dos hay un tercer grupo.

Es decir, por ejemplo, cuando el grupo 1(puesto de control), da orden de que el coche avance, la orden será enviada al cerebro y éste dirá al cuerpo que mueva el motor. Nunca se hará directamente grupo 1 – grupo 3.

Tampoco es posible que el cuerpo le dé una orden al cerebro, ni que el cerebro efectúe operaciones sobre el puesto de control. Toda orden siempre viene de un nivel superior.

Estableciendo el sistema de esta manera tenemos la certeza de que siempre el usuario tendrá el control absoluto del vehículo. En ningún momento una orden del usuario será ignorada ni se dudará de si ejecutarla o no.

El grupo uno se comunicará con el dos y a su vez recibirá información de éste (nunca órdenes), por medio de una comunicación inalámbrica, esta será Wi-Fi, tal y como se especifica en los requisitos de diseño.

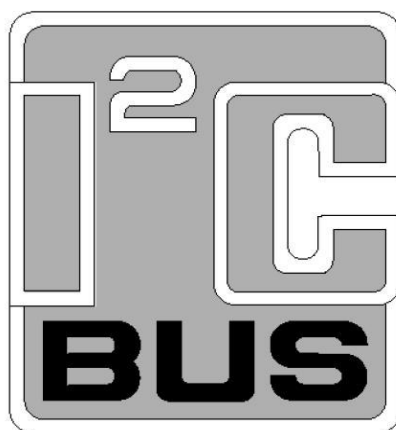
Mediante el empleo de un Router se creará la red privada usada para conectar el puesto de control (grupo 1) y el cerebro (grupo 2). Esta red estará encriptada y tendrá clave de seguridad para que nadie más que el usuario pueda acceder a ella y tomar el control del sistema.

La decisión de haber usado esta red es debida al gran uso que tiene hoy en día a nivel mundial y a la facilidad de trabajar con ella.

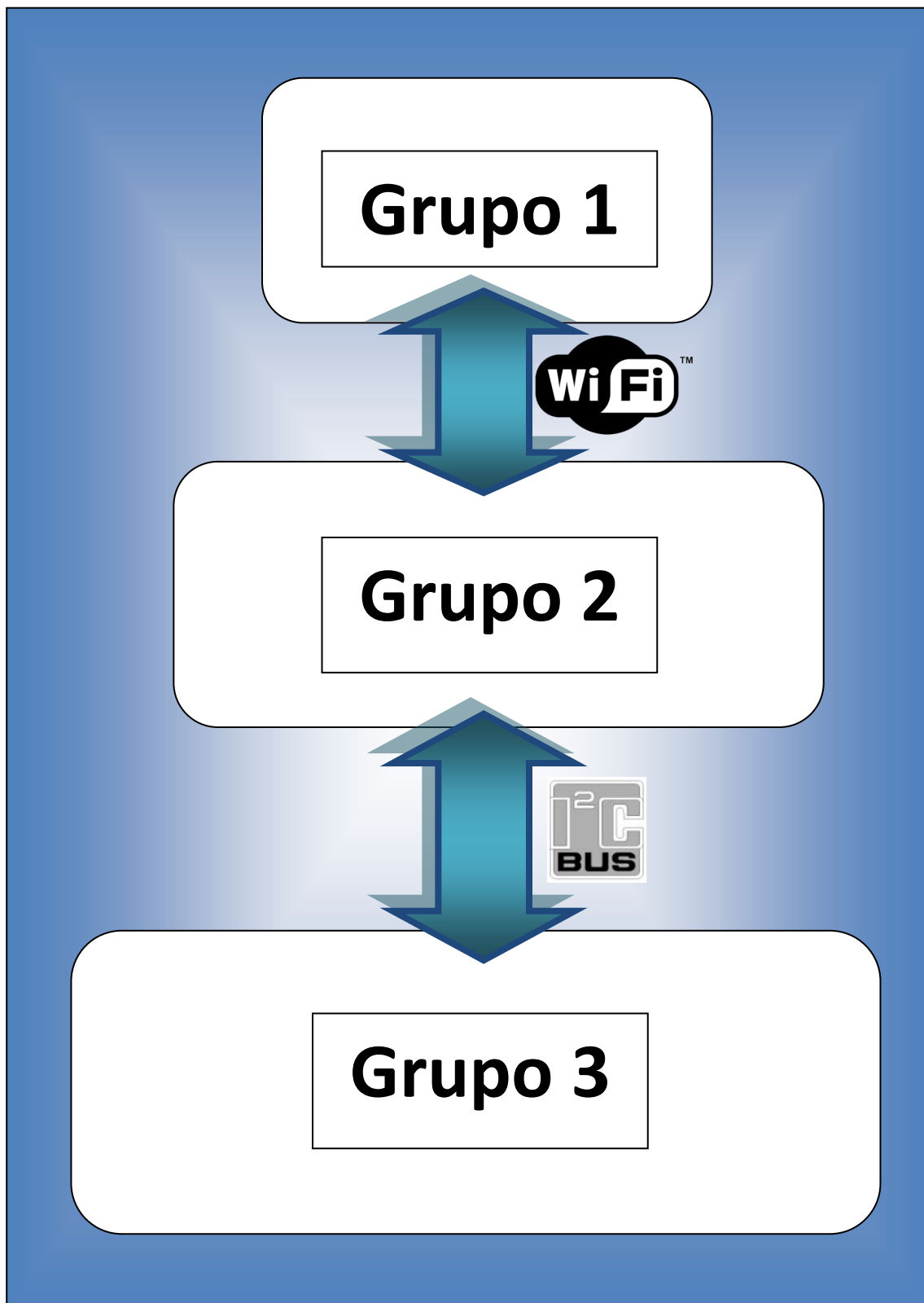


**Figura 56:** logo Wi-Fi

La comunicación del grupo dos con el grupo 3, el cerebro con el cuerpo, se ha hecho mediante un único bus de comunicación I2C. Aunque la comunicación SPI es más rápida que el I2C, no se sabía de su manejo ni se tenía experiencia alguna con ella, con la comunicación I2C sí, se había trabajado anteriormente con ella y se poseía experiencia.



**Figura 57:** Logo comunicación I2C



**Figura 58:** esquema general del sistema

Si definimos el sistema desde un punto de vista físico, se compondría de dos partes. Una sería el robot, el vehículo en sí, en donde se encontrarían los grupos 2 y 3. La otra parte sería el ordenador, desde donde se gobierna todo entero, estando aquí el grupo 1.



**Figura 59:** aspecto físico del sistema

Se describen a continuación cada uno de estos grupos, de que se componen, como funciona y que tareas se llevan a cabo en ellos. Se explica tanto la parte hardware como la parte software de los mismos. Incluyendo un esquema de cada uno para así mostrar de una forma más visual su manera de proceder.



### 1.8.3.2. Grupo 1

#### 1.8.3.2.1. Descripción general

El grupo uno constituye todo el puesto de control del sistema. Desde él, se pueden emitir órdenes al vehículo, tales como movimiento, controlar el motor de aceleración, la articulación, etc. Se estarán recibiendo las imágenes del dispositivo de captura que se emplee. Es desde aquí desde donde se le dará el orden de iniciar la vigilancia. La tarea de este grupo es la de controlar al vehículo en base a lo deseado por el usuario.

##### 1.8.3.2.1.1. Composición

El grupo se compone de:

- Ordenador
- Router
- Joystick Analógico

##### 1.8.3.2.1.2. Esquema



**Figura 60:** Esquema grupo 1

### 1.8.3.2.2. Funcionamiento global

El funcionamiento es bastante sencillo: en el ordenador habrá un software, que el usuario emplea para dar las órdenes.

El vehículo realizará la acción que el usuario desee, si se quiere que el coche funcione solo, habrá que darle al botón correspondiente. Si se desea asumir el control del vehículo, no habrá más que desactivarlo y empezar a manejarlo con los distintos botones de pantalla, teclado y joystick.

Antes de poder usar el software la persona que vaya a usarlo deberá registrarse dando su nombre y una contraseña. Esto se hace por motivos de seguridad, para saber quién estaba manejando el vehículo o sobre quién está la responsabilidad en cada momento de lo que ocurra.

Sin duda este grupo no ha precisado ninguna parte de montaje, puesto que es todo software, el único hardware que ha tenido que trabajarse un poco ha sido el joystick. Más adelante se explica el motivo.

### 1.8.3.2.3. Funcionamiento individual

#### 1.8.3.2.3.1. Ordenador

El ordenador usado para el puesto de control ha sido el ordenador personal disponible. Un ordenador Modelo HP Pavilion G6, intel CORE I5, aunque cualquier ordenador similar valdría para esta aplicación.

Para llevar a cabo el desarrollo de la interfaz de usuario, desde la cual se iba a controlar el robot, se decidió escribirla desde LinuxMint 12, una distribución de Linux profesional pensado para el desarrollo de aplicaciones informáticas.

El ordenador disponía de conexión Wi-Fi, con lo que no había que acoplarle ningún adaptador de red para que se pudiese conectar.



**Figura 61:** Ordenador portátil

El hecho de que se haya elegido un ordenador portátil, es que si se cogiese uno de sobremesa, siempre deberá de estar en el mismo sitio, si queremos buscar flexibilidad a la hora de moverse, esa no es la opción correcta, por lo que un portátil es lo adecuado. En el ordenador estará funcionando la interfaz de usuario diseñada para la aplicación. Desde la que se podrá tener control de todo el sistema. El software de diseño y como se ha llevado a cabo se explica en el apartado software.

#### **1.8.3.2.3.2. Router**

Se decidió usar una red Wi-Fi privada, en la que pudiésemos trabajar sin problemas de sobrecarga debido a otros usuarios. Para elegir el Router no se pensó en ninguno en particular ni se siguió ninguna pauta debido a no tener los conocimientos adecuados para ello. Simplemente se buscó uno disponible y operativo, se tomó un Router de Vodafone, el cual se reseteó y se configuró al gusto. Tras comprobar que funcionaba correctamente se tomó como el elegido. Para el propósito que se persigue valdría cualquier router que pudiese generar su propia red.



**Figura 62:** Router utilizado

#### **1.8.3.2.3.3. Joystick**

Dado que un requisito es el poder controlar el coche a gusto del usuario, se pensó en una forma cómoda de hacer, así que se buscó algún dispositivo el cual pudiese ayudar a realizar esta tarea, un dispositivo pensado exclusivamente para el control manual del vehículo.

Se pensó en un volante, pero debido a su tamaño y al coste se descartó. La elección fue un joystick analógico.

El joystick elegido debía de ser analógico y disponer de suficientes botones para las funciones que se quisiesen realizar, el modelo elegido es el Joystick DB Boeder. Un joystick profesional y ergonómico.

Este dispone de las siguientes características:

- 2 ejes analógicos.
- Hat o POV.
- 4 botones.
- 1 acelerador.



**Figura 63:** Joystick

El controlar un vehículo con este elemento, se hace de la siguiente forma. Para que se ande hacia delante o hacia detrás, bastará con inclinar respectivamente el joystick, hacia delante o hacia detrás. Como es analógico, cuanto más se incline más velocidad se dará al vehículo y contra menos, pues más despacio irá.

Controlar la dirección sigue el mismo principio, para girar a la derecha o a la izquierda también se deberá inclinarlo, aquí si se inclina más o menos, pues girará más o menos. Usando el hat se controlará la articulación que hubiese.

Esta es una forma cómoda de manejar el robot. Cabe también la posibilidad de hacerlo directamente desde el teclado del ordenador, pero resulta mucho más incómodo y se pierde la proporcionalidad. No se puede dar más velocidad o menos, siempre constante.

Había un inconveniente con el joystick y es que dispone de una conexión de puerto de juego de 15 pines. Esta conexión se quedó obsoleta cuando apareció la comunicación USB. Con lo que hubo que buscar alguna forma de poder conectarlo.

Se solucionó con un adaptador de puerto de juego a USB disponible en internet.



**Figura 64:** adaptador Super Joy Box 7

Este adaptador fue la perfecta solución ya que el ordenador lo detectaba como si de un joystick normal se tratase, no necesitaba alimentación y tampoco se precisó de ningún driver extra para poder usar el joystick con normalidad, fue el sistema operativo el que lo detectó y lo configuró.

Lo único que se debía saber es que el adaptador viene preparado para tres dispositivos de juego concretos, un joystick, un volante o un GamePad, habiendo que configurarlo para usar cada uno de estos. La forma de configurarlo es mediante un interruptor incorporado. Además, se observa a simple vista que no dispone de ningún tipo de botón extra ni pulsador.

Para usarlo correctamente tendremos que prepararlo para trabajar con el dispositivo que queremos, en nuestro caso es un joystick. La forma de proceder es poner el interruptor que trae en la posición correcta.

Se observará que son tres letras impresas sobre el interruptor, como se puede observar en la figura 1.24, las letras son: P, W y S. Cada letra se corresponde con uno de los dispositivos, que se le pueden conectar, de la siguiente forma:

- P -> Pad, este es la posición de GamePad.
- W -> Wheel, rueda, es aquí donde se pondrá si se quiere conectar un volante.
- S -> Stick, en esta posición deberá ponerse para indicar un joystick.

Una vez bien configurado y conectado, se puede leer correctamente la información proveniente del joystick mediante una cómoda conexión USB.

#### 1.8.3.2.3.4. Software

Para llevar a cabo esta tarea, se ha usado el entorno de desarrollo QT Creator.

Qt Creator es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt, requiriendo su versión 4.x. Actualmente la compañía es propiedad de Nokia.



**Figura 65:** Logo QT Creator

Al usar este entorno estamos ante un entorno de desarrollo con una amplia serie de librerías para el desarrollo de aplicaciones gráficas. Todas son de fácil acceso a parte de ser free software siempre que no se requieran fines económicos.

Existe una gran cantidad de usuarios de QT en todo el mundo , una gran ayuda a parte de miles de foros en los que se solventan todas las dudas que surgen. Gracias a éstos se han solucionado muchos problemas que fueron surgiendo.

### 1.8.3.3. Grupo 2

#### 1.8.3.3.1. Descripción general

El grupo dos tiene dos tareas fundamentales. Una es obedecer las órdenes que lleguen del grupo 1 sin dilación. Nunca se cuestionará una orden ni se ignorará. La segunda es tomar las decisiones cuando el control del vehículo lo tenga este grupo. De ahí que el apodo sea “El cerebro”.

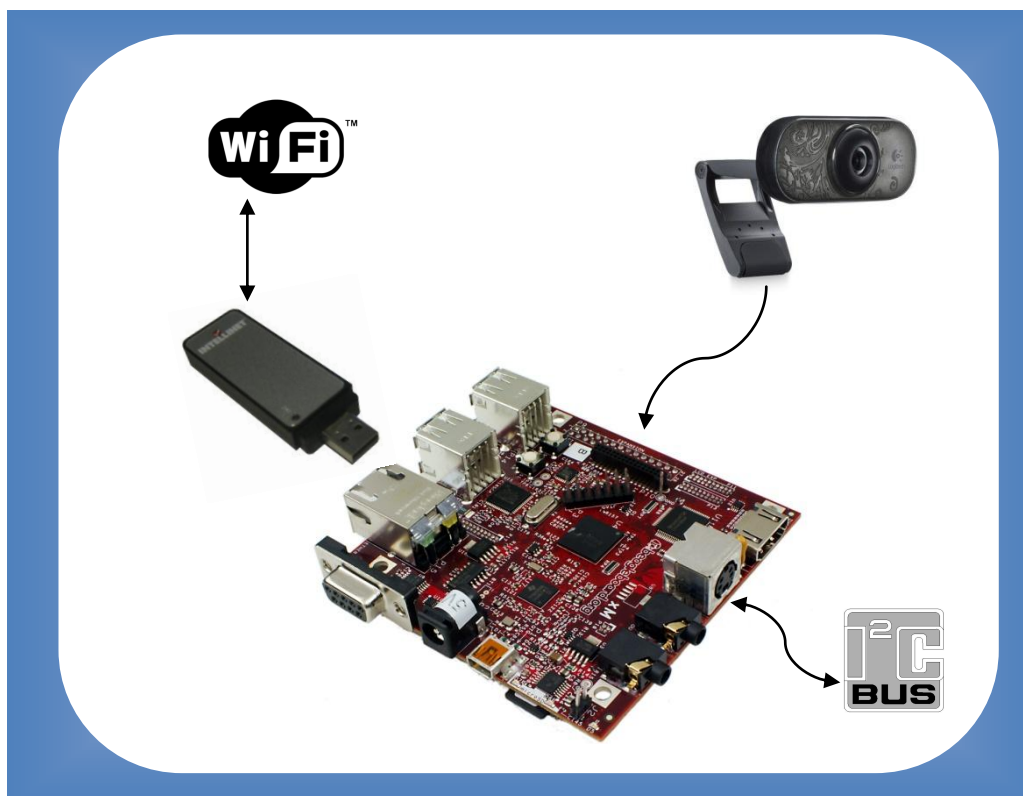
Este grupo está dotado de un potente procesador para llevar a cabo la sublime tarea que se le encomienda.

##### 1.8.3.3.1.1. Composición

El grupo se compone de los siguientes elementos:

- Sistema embebido BeagleBoard-xM
- Webcam Logitech c-210
- Adaptador Wi-Fi USB.

##### 1.8.3.3.1.2. Esquema



**Figura 66:** Esquema grupo 2

### 1.8.3.3.2. Funcionamiento global

Este grupo recibirá las órdenes del grupo 1, es decir, del usuario. Estas deberán ser escuchadas y ejecutadas en el momento. No habrá dilación ni cuestión.

Las órdenes entrantes pueden ser de dos tipos. El primero es de control del vehículo, esto significa que la información entrante contiene instrucciones para realizar una acción determinada, por ejemplo, mover la articulación, acelerar el motor, etc. El segundo tipo sirve para comunicar al coche el inicio de la vigilancia, cuando se dice vigilar, se refiere a seguir un camino no establecido, evitando obstáculos e intentando encontrar perturbaciones en el entorno. Cuando el coche esté funcionando así, es él mismo el que debe de tomar decisiones respecto a su movimiento, éstas se apoyarán en la información de los sensores, en el estado actual del vehículo y en las imágenes captadas por la cámara.

El proceso que seguirá para tomar una decisión es el siguiente.

1. Interpretación de la imagen.
2. Interpretación de los sensores
3. Tener en cuenta con situación pasada y actual
4. Evaluar.

Toda la información de los sensores se recibirá a través de la comunicación con el grupo 3, que se realiza por I2C. Las acciones de movimiento del vehículo, se enviarán también por este mismo bus.

Las órdenes que lleguen del grupo 1, así como la información que se envíe a este, lo harán por una red Wi-Fi.

El centro de procesamiento, en donde se llevará a cabo el grueso de la programación, lo constituye un sistema embebido denominado BeagleBoard-xM.

El sistema operativo Linux ha sido sin duda la mejor opción para llevar a cabo todo lo que se le encomienda.

La cámara elegida ha sido una Webcam conectada por USB, debido a la comodidad de esta conexión y a la flexibilidad respecto a la distancia de colocación dado a la longitud del cable de conexión. Se recogerán permanentemente imágenes de ella.

Mientras se lleva a cabo esto, se están realizando otras tareas de manera continua, como enviar la imagen recibida de la cámara al grupo 1, es decir al ordenador, para poder controlar en todo momento la visión del vehículo. Más adelante se explica cómo se realiza esto.



### 1.8.3.3.3. Funcionamiento de cada elemento

#### 1.8.3.3.3.1. Webcam

Al final se decidió por una webcam debido a la cómoda conexión por USB. La Webcam elegida es la cámara Logitech c210. El motivo por el que se decidió este modelo es por experiencias personales de otros usuarios de la misma placa elegida para el vehículo, la BeagleBoard-xM. Otro gran motivo para decantarse por esta fue que al estar conectado por USB esto permitía una longitud de cable considerable entre la BeagleBoard-xM y la cabeza articulada, con lo cual, se podía situar a la cámara en el lugar deseado.

La cámara puede obtener imágenes de hasta 1,3 megapíxeles, pero nosotros utilizaremos una resolución de 320x240. Esta resolución es bastante reducida pero es necesario hacerlo así debido a que vamos a estar continuamente captando imágenes, procesándolas y a su vez enviándolas a través de una red Wi-Fi.



**Figura 67:** cámara Logitech c210

#### 1.8.3.3.3.2. Adaptador Wi-Fi USB

Para poder conectar todo el vehículo a una red inalámbrica Wi-Fi se han recurrido a los puertos USB disponibles en la BeagleBoard-xM. Se ha usado un adaptador de red inalámbrica. La única función que tiene es la de poder conectar el grupo 2 con el grupo 1 a través de la red. Para elegir el modelo no ha habido que buscar demasiado, el que se poseía se probó y Linux automáticamente lo instaló, así quedó solucionada la comunicación inalámbrica por Wi-Fi.



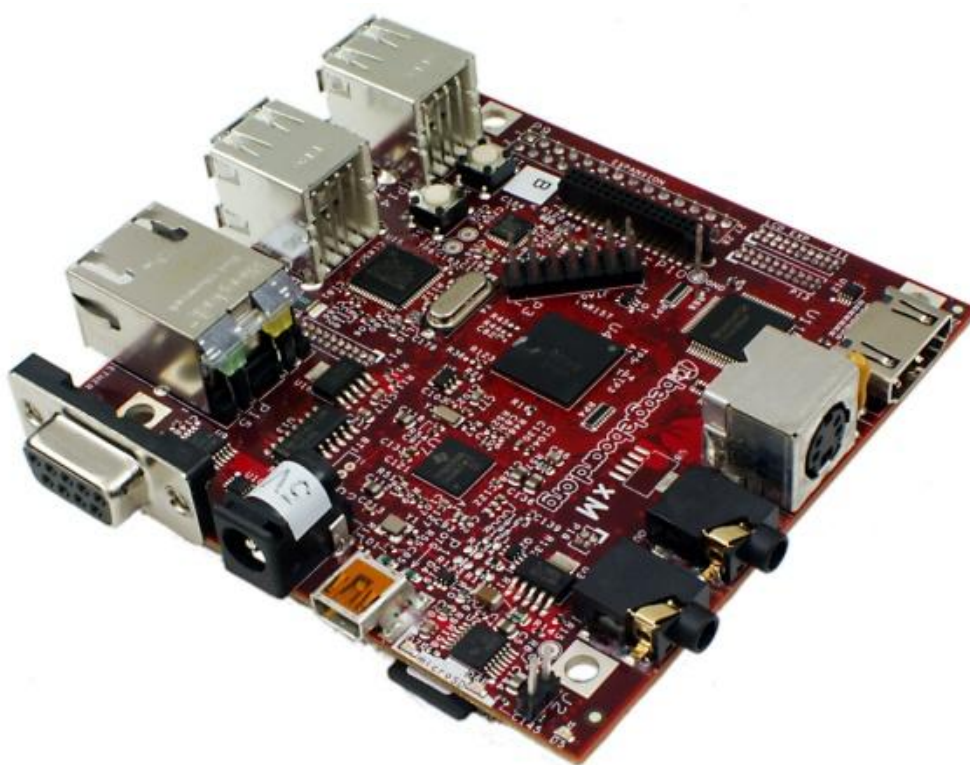
**Figura 68:** Adaptador Wi-Fi USB

### 1.8.3.3.3. BeagleBoard-xM

La BeagleBoard-xM es el centro de procesamiento de todo el vehículo.

Se trata de un sistema embebido desarrollado por los ingenieros de BeagleBoard. Esta placa es la respuesta a aquellas personas que necesitaban un soporte potente para llevar a cabo aplicaciones que requerían de una mayor capacidad de procesamiento sin tener que usar un ordenador de por medio.

Esta placa tiene la potencia suficiente para ejecutar el sistema operativo Linux. La distribución más usada en ella Angstrom aunque también es posible instalar otras. Dispone de 4 puertos USB, Ethernet, SPI, I2C, RS232 y más. Posee un procesador ARM<sup>®</sup> Cortex<sup>™</sup> -A8 a 1 GHz y 512 MB de memoria RAM DDR de bajo consumo.



**Figura 69:** BeagleBoard-xM

Su misión engloba casi toda la tarea que tiene el grupo 2.

Su trabajo consiste en estar continuamente haciendo lo siguiente:

- Recoger imágenes de la cámara y mandándolas por Wi-Fi
- Esperar órdenes del grupo 1 para llevarlas a cabo.
- Cuando se le ordene, tomar el control del vehículo e iniciar el programa de vigilancia, tomando las decisiones que crea oportunas.

### 1.8.3.3.4. Software

#### 1.8.3.3.4.1. Funcionamiento general

La Beagleboard-xM es el centro neurálgico del vehículo. El software que se estará ejecutando en ella es capaz de recibir órdenes por parte del usuario, entre las cuales se encuentra la orden de iniciar la vigilancia.

Se trata de un software que permanece constantemente captando imágenes, seguido del envío de las mismas al ordenador, actividad que desarrollará a pesar de no existir entrada alguna de órdenes.

En el instante en el que se produce la entrada de una orden, ésta será ejecutada por el software, continuando la salida de imágenes hacia el ordenador. Si la orden que se produce es la de inicio de vigilancia, entonces el programa comenzará el procesamiento de las imágenes, función que desempeñará paralelamente a la transmisión de las imágenes.

Una vez que la orden de vigilancia se encuentra activada el software gobernará y tomará decisiones de movimiento del vehículo.

#### 1.8.3.3.4.2. Programación principal

La programación multihilo consiste en que un programa sea capaz de ejecutar varias tareas de forma paralela, realmente este paralelismo no es del todo cierto, ya que lo que verdaderamente ocurre es que se reparte la ejecución entre dos procesos de forma que ocurran casi a la par.

Para lograr esta programación multihilo se han usado las librerías GStreamer. Estas librerías incorporan elementos preparados para trabajar con contenido multimedia, permitiendo hacer streaming de video y de audio.



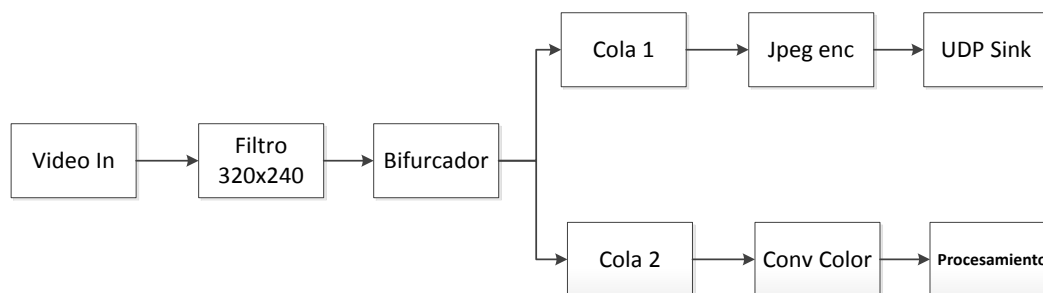
Figura 70: Logo librerías gstreamer

Su funcionamiento se basa en lo que se denominan “pipelines”. Un pipeline consiste de una serie de elementos enlazados entre sí, teniendo cada elemento una función particular. Cada elemento recibe información del anterior, realiza una determinada transformación y pasa el resultado al siguiente (o siguientes) elemento del pipeline. Los elementos se enlazan entre sí para llevar a cabo la tarea global requerida.

La biblioteca GStreamer puede usarse para trabajar tanto con video como con sonido, habiendo codificadores, decodificadores, sumideros, fuentes, etc. para llevar a cabo distintas funciones y diferentes tipos de fuentes y dispositivos de salida.

Una aplicación que use GStreamer puede crear varios pipelines para realizar “simultáneamente” varias tareas de procesamiento de audio o vídeo. Cada pipeline creado se comportará como un hilo independiente dentro de la aplicación.

El pipeline creado para esta aplicación es el siguiente:



**Figura 71:** Pipeline seguido

Como se aprecia, el pipeline dispone de una bifurcación para conseguir enviar datos tanto al grupo 1, a través del enlace por Wi-Fi utilizando protocolo UDP, como a las funciones que se ocuparán del procesamiento de las imágenes dentro del mismo software en la Beagleboard-xM.

A continuación se detalla cada elemento de que consta el pipeline:

El primer elemento es de un tipo denominado “Video-Input”, este crea una puerta de entrada de imágenes asociada a algún dispositivo disponible para ello, es decir, una cámara conectada o alguna conexión a alguna cámara remota. Se trata de una fuente de información.

El segundo elemento, es un filtro, que lo que hace es, captar las imágenes de la cámara según las dimensiones y velocidad que decidamos, en nuestro caso la resolución se ha establecido en 320x240 píxeles, tamaño adecuado para poder ser procesadas y enviadas sin sobrecargar el sistema.

Para la velocidad de adquisición, teniendo en cuenta que hay un procesamiento y un envío de información a través de una red inalámbrica, esta se ha establecido en 15 frames por segundo.

El Bifurcador y las 2 colas son los elementos que crean los caminos paralelos, vienen incluidos en la librería gstreamer. Con estos tres elementos se pueden distribuir datos a otros elementos a la vez, aquí sólo se hace a dos, de esta forma, logramos que se pueda, enviar la imagen a través de Wi-Fi y que a su vez sea enviada a una función para que sea procesada al mismo tiempo.

En la bifurcación superior en el esquema, el elemento “jpeg enc” lo que realiza es una compresión de cada imagen tomada. Dejándolo listo para cedérselo al siguiente elemento de la cadena.

El último elemento del camino superior es el “UDP sink”, este elemento realiza una emisión de datos a través de una red inalámbrica. Lo realiza por UDP, habiéndole que especificar el puerto usado para ello y la IP del equipo de destino. Este protocolo de comunicación no es elegido al azar, pudiendo haberse también el protocolo TCP. El motivo es que el video que se manda por red inalámbrica al ordenador, es decir, al grupo 1 es meramente informativo para el usuario que está usando el sistema. No hay ningún proceso en el ordenador que dependa de las imágenes, ni se le realiza ningún tratamiento de ningún tipo, solamente el mostrarlas en pantalla. Por lo que al ser UDP no precisa de acuse de recibo para verificar que la imagen llegó al ordenador, así que si algún frame no llega, realmente no provoca ningún problema al sistema, salvo al observador de la pantalla, que apreciará en pantalla lo que se denomina un “salto”.

En la bifurcación inferior del esquema, hay un elemento que se llama conversor de color, este elemento está debido a que el procesamiento de las imágenes trabaja con el espacio de colores BGR y no tal y como vienen de la cámara RGB.

Una vez hecho esto pasamos al último elemento. Se trata de un sumidero, en el cual llegan las imágenes y pasan a una función propia en la que estas se procesaran.

Este pipeline, se estará ejecutando de manera continua, por lo que en el momento que se inicie la ejecución del software, se estarán enviando imágenes por UDP y procesándolas al mismo tiempo.

#### 1.8.3.3.4.3. Procesamiento de imágenes

Para procesar las imágenes se empleará las librerías OpenCV.



**Figura 72:** Logo de OpenCV

OpenCV es una librería de Visión Artificial de Software libre. Éstas permiten el procesamiento de las imágenes de manera fácil y sencilla, dejando de lado las matemáticas y centrándose únicamente en la operación que se quiera hacer con dichas imágenes.

En el último elemento de la bifurcación inferior del pipeline, el que tiene por nombre “Procesamiento”, es en donde se llevará a cabo el tratamiento de cada imagen, función condicionada a que se esté en modo de vigilancia, si esto se cumple, con la información que se saque del procesamiento, más la entrante de los sensores, se tomarán las decisiones oportunas.

#### **1.8.3.3.4.4. Autonomía**

Cuando se le dé la orden de vigilancia, será el vehículo el que se gobierne a sí mismo. Para la toma de decisiones se seguirán instrucciones dadas previamente.

Es el usuario, el que puede decidir qué acciones llevar a cabo y en base a qué, ya que se puede preparar para trabajar en distintas situaciones.

La lógica que se le ha implementado al vehículo, es la de continuar una pared, permaneciendo a una distancia prudencial de ella pudiendo evitar algún obstáculo que se encuentre delante, realizando la maniobra de rectificación de la trayectoria.

Debido a la dificultad del procesamiento de imágenes, se ha realizado una función para que detecte círculos en el procesamiento de las imágenes, cuando esto ocurra, será comunicado al grupo 1.

#### **1.8.3.3.4.5. Comunicación con grupo uno**

Las órdenes vendrán del grupo 1 por Wi-Fi, por lo tanto hay que tener algún método de comunicarse a través de este tipo de red inalámbrica. Usaremos lo que se denominan sockets.

Un socket es un objeto abstracto que permite el intercambio de datos entre dos programas a través de una red, estando normalmente los programas en computadoras distintas. Usándolo se crea un canal para poder transmitir y recibir datos entre la Beagleboard-xM y el ordenador.

Los sockets funcionan por eventos, es decir, cuando se envíe algún dato del ordenador (grupo 1) a la Beagleboard-xM (grupo 2) se generará un evento anunciando que se hay datos disponibles para ser leídos, lo que denominamos una orden. Los datos se leerán y se continuará con la ejecución normal del programa siguiendo el pipeline creado.

Para interpretar los datos se ha creado una serie de códigos, que según sea uno u otro se hará una cosa u otra, por ejemplo, mover el motor tiene una serie de códigos, ya sea acelerar, desacelerar etc, y mover un servomotor determinado tiene otra serie de códigos.

Es decir, hemos creado, que cuando haya datos disponibles para ser leídos, se genere una interrupción, se lea la información se interprete y se continúe normalmente con el programa principal.

#### **1.8.3.3.4.6. Comunicación con grupo tres**

Para comunicar al grupo 3 las funciones de movimiento que se tengan que hacer o recibir información de algún sensor se hará mediante un bus de comunicación i2c. Para ello, se ha recurrido a un archivo disponible en la Beagleboard-xM que sirve de manejador para el bus disponible en la placa. Primeramente habrá que abrirlo por software, una vez abierto, cada vez que se escriba información en él se enviará por el bus, a su vez cuando se lea información significa que se han recibido datos.

Se han creado funciones para actuar sobre diferentes elementos del grupo 3 o para recibir información de sensores, cuando se llamen a éstas se enviará por i2c una serie de códigos que indicarán la acción requerida.

### 1.8.3.4. Grupo 3

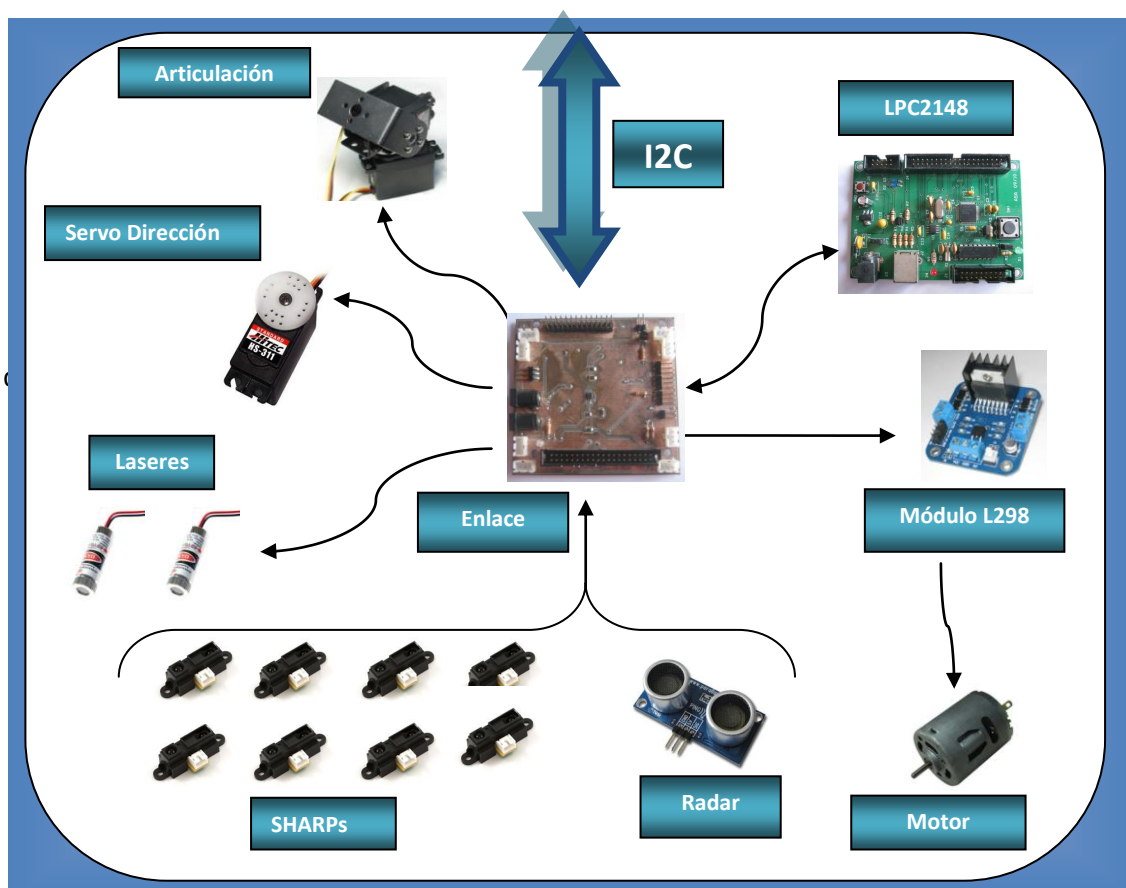
#### 1.8.3.4.1. Descripción general

El grupo tres es el que se encuentra más abajo en la jerarquía del sistema. Es el encargado de todas las funciones motrices, de ahí su apodo “El cuerpo”. Recibe órdenes del grupo dos. En base a estas órdenes así será su comportamiento.

##### 1.8.3.4.1.1. Composición

- Placa con Microcontrolador LPC2148.
- Tres Servomotores.
- Motor de corriente continua.
- Módulo L298.
- Radar.
- Ocho sensores de proximidad por infrarrojos.
- Dos Láseres rojos.
- Placa de enlace.

##### 1.8.3.4.1.2. Esquema



**Figura 73:** Esquema grupo 3

### **1.8.3.4.2. Funcionamiento global**

El centro de control de este grupo es el microcontrolador LPC2148. Es a él a quien le van a llegar las órdenes procedentes del grupo dos. Dado que la comunicación es I2C, esta implica un maestro y un esclavo. El micro se ha configurado como esclavo constante, cumpliendo así las leyes de la jerarquía del sistema. Estando a la escucha permanentemente de órdenes entrantes.

Mientras que esté esperando órdenes, no está sin trabajo. Está constantemente capturando el valor de los sensores y almacenándolos en una pila. Así como manteniendo el valor que tengan en ese momento los servos y motor.

#### **1.8.3.4.2.1. Órdenes**

Las órdenes entrantes pueden ser de dos tipos:

- De ejecución
- De devolución

##### **1.8.3.4.2.1.1. Orden de ejecución**

Esta ordena al microcontrolador una función a llevar a cabo. La función va a tener como objetivo el trabajar sobre uno de los servomotores, sobre el motor o los láseres, en el caso del servo modificar su posición, en el caso del motor su aceleración y sentido, en el caso de los láseres encenderlos o apagarlos.

No hay retorno. Cuando la orden se ha ejecutado el microcontrolador sigue a la espera de nuevas órdenes.

##### **1.8.3.4.2.1.2. Orden de devolución**

Este tipo de orden es el que implica un retorno en la comunicación. El mandato que trae es el deseo de querer conocer el valor de uno de los sensores almacenados en buffer. La orden en sí indicará el sensor que se desea conocer, inmediatamente después el valor de este será mandado por el bus.

Sobre este grupo se sostienen todos los demás, si fallase, todo el sistema quedaría estancado. Ya que por mucho que se insista, en mover un servo (por ejemplo), si este grupo no es capaz de hacerlo, no se podrá llevar a cabo.

El hecho de haber elegido el microcontrolador como encargado de toda esta tarea, es debido a experiencia en él, habiendo trabajado lo suficiente para saber manejarlo sin problemas. Además de venir preparado para este tipo de tareas, antes de incluirlo, se han hecho innumerables pruebas individuales con cada elemento a conectar en el vehículo. Se ha probado que todo funcionaba antes de ensamblarlo.



### 1.8.3.4.3. Funcionamiento de cada elemento o elementos

#### 1.8.3.4.3.1. El microcontrolador

El microcontrolador elegido es el NXP LPC2148. Su elección es debido a su uso en la asignatura “Aplicaciones de Sistemas y Automática”, en donde se llevó a cabo el diseño y desarrollo de una placa que funcionase en base a este integrado.

Se diseñó la placa con ORCAD, se mandó fabricar y se soldó posteriormente.

El microcontrolador está basado en arquitectura ARM7TDMI de 32 bits. Con una memoria flash de 512kB.

Se comunica con el grupo superior a través de un canal de comunicación I2C, estando configurado permanentemente como esclavo a la espera de órdenes.

Se ha programado en C++ usando el entorno de desarrollo CrossWorks de Rowley Associates.

Su misión es la de controlar la posición de los servomotores, la aceleración y sentido del motor, encender y apagar los láseres y llevar a cabo la lectura de todos los sensores del coche, almacenando su valor en un buffer.

El microcontrolador está implementado en una placa de desarrollo fabricada en la asignatura de “Aplicaciones de Sistemas y Automática”.



Figura 74: placa con microcontrolador LPC2148

El software que lleva el microcontrolador se ha desarrollado en un sistema operativo Windows 7, utilizando para desarrollarlo el sistema “CrossWorks for ARM.” de la empresa Rowley Associates.

En sí el entorno se denomina CrossStudio. Preparado para trabajar con microcontroladores y microprocesadores con arquitectura interna ARM tiene una opción para trabajar con nuestro modelo, el LPC2148.

En él se puede desarrollar código en lenguaje de programación C/C++. Viene con su propio compilador. Una vez elegido y creado el proyecto a realizar se puede empezar a programar.

Cuando se programa un micro con este entorno, se pueden realizar dos acciones para comprobarlo. Una es, usar el simulador de ARM que incorpora el programa, eligiendo como opción de destino el nombrado ARM Simulator. La otra es volcar el programa en nuestro microcontrolador, para ello, una vez escrito el código y compilado, usaremos el depurador para poder volcarlo en la placa y ejecutarlo. Para ello necesitaremos de una conexión, que se realiza mediante un depurador.

Para realizar el volcado en la placa, hemos usado el debugger JTAG ARM USB TINY.



**Figura 75:** depurador JTAG ARM USB Tiny

Una vez que teníamos el conjunto implementado se puede desarrollar el programa.

Este depurador junto con el entorno CrossWorks no solo permite el programar el micro, sino que también permite controlar el avance del programa, controlar variables, seguirlas, controlar las interrupciones, es decir, un control total.

Cuando se comprueba que el programa funciona correctamente se puede preparar el micro para que automáticamente lo inicie cuando se encienda.

Más adelante se detalla el comportamiento del software del microcontrolador.

### 1.8.3.4.3.2. Sensores

El vehículo dispone de nueve sensores:

- Un radar por ultrasonidos.
- Ocho sensores de distancia infrarrojos.

#### 1.8.3.4.3.2.1. Sensores Infrarrojos

Debido a la necesidad de conocer la distancia respecto a los objetos del entorno se decidió el empleo de sensores de distancia. El uso de los infrarrojos es porque miden la distancia en línea recta.

Su respuesta es analógica, así que para medirlos se usaran los convertidores A/D del microcontrolador.

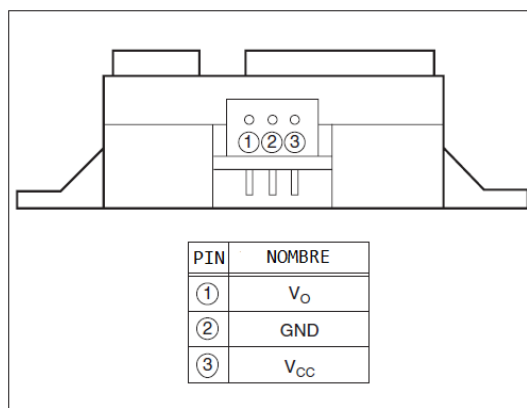
En el mercado existen una gran variedad de rangos de medida para este tipo de sensores, pero el elegido fue de 10 a 80 centímetros, en concreto el modelo SHARP GP2D12. Cuya distancia es suficiente para saber si hay algún objeto alrededor y si va a haber una colisión.



**Figura 76:** sensor infrarrojo

#### 1.8.3.4.3.2.1.1. Conexión

El sensor dispone de tres conexiones como se muestra en la imagen:



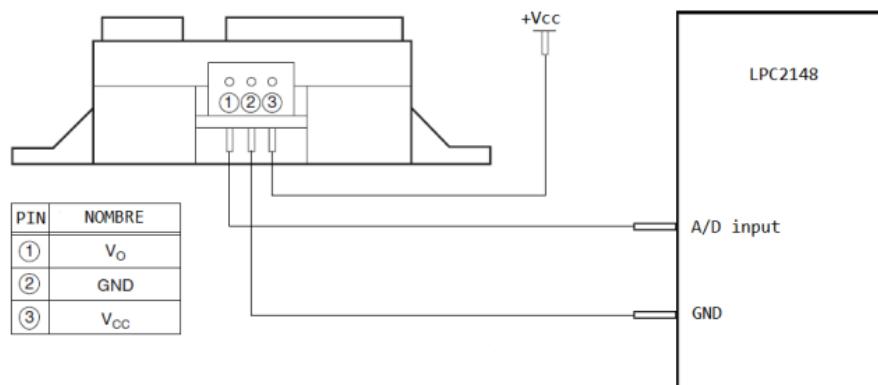
**Figura 77:** pines sensor infrarrojo

A continuación se detalla cada una:

- Vcc: Tensión de alimentación del sensor, según el datasheet, 5V.
- GND: tierra.
- Vo: Tensión de salida, es esta la que se leerá para saber el valor de la distancia del sensor en ese momento.

Para llevar a cabo su lectura se lleva a cabo el siguiente procedimiento:

- 1°. Vcc se conecta a la tensión de alimentación de los sensores, en este caso 5V.
- 2°. Vo se conecta a una entrada analógica del microcontrolador. Esta entrada estará configurada como entrada analógica, esto se hace mediante programación.
- 3°. GND se conecta a GND de alimentación debiendo de estar el microcontrolador también conectado a esta GND para que la lectura pueda ser válida.



**Figura 78:** conexión del sensor infrarrojo

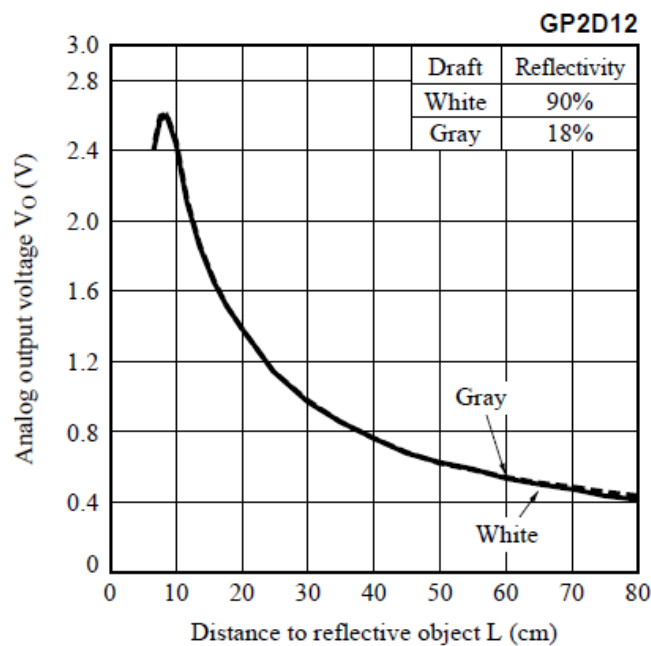
### 1.8.3.4.3.2.1.2. Funcionamiento

El funcionamiento del sensor es bastante sencillo, en base a la distancia a la que esté el objeto así será su salida. Teniendo muy en cuenta los rangos de medida. Este sensor en concreto dará una salida fiable si la distancia al objeto medido está en el rango 10-80cm.

Si el objeto está fuera de rango el sensor dará una salida, que será totalmente legible, pero que no tendrá nada que ver con la realidad del entorno.

### 1.8.3.4.3.2.1.3. Salida

La salida del sensor, es decir, la tensión que hay en  $V_o$ , respecto a GND varía en función de la distancia a la que este el objeto del sensor dentro del rango. Este en concreto sigue la siguiente grafica Voltaje-Distancia:



**Figura 79:** Gráfica de salida del sensor infrarrojo

### 1.8.3.4.3.2.2. Radar ultrasónico

Se quiso disponer también un sensor de distancia de largo alcance, con una salida fiable. La solución fue el radar. El modelo Parallax PING)).

Este dispositivo permite captar un elemento hasta una distancia de tres metros. El inconveniente que tiene es que funciona en cono, con una abertura de  $20^\circ$  de eje central a eje de revolución, es decir,  $40^\circ$  en la punta de sección.

El dispositivo dará la misma salida, estando el objeto en un arco del cono. Pero como va a la par que la cámara, detectando la distancia se podrá ver en la imagen la posición del objeto.



**Figura 80:** radar ultrasónico

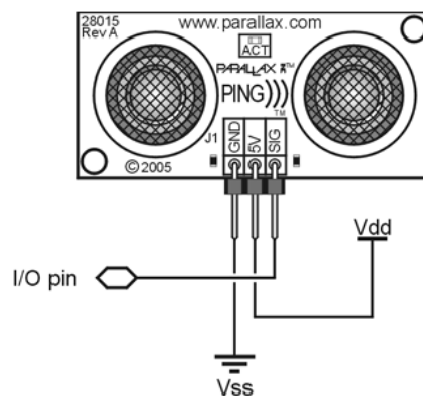
Es de vital importancia el conocimiento de la zona en la que se utilizará, ya que funciona por eco, (más adelante se detalla su funcionamiento).

Si en la zona en la que se usa se emiten vibraciones que pudiesen provocar la emisión de ultrasonidos estos podrían en conflicto con el sensor, recibiendo un eco que no fuese el emitido por este.

Este sensor permitirá detectar objetos que se encuentren en frente del vehículo a una distancia considerable.

#### 1.8.3.4.3.2.2.1. Conexión

Este sensor dispone de tres conexiones como se muestra en la imagen:



**Figura 81:** pines del radar

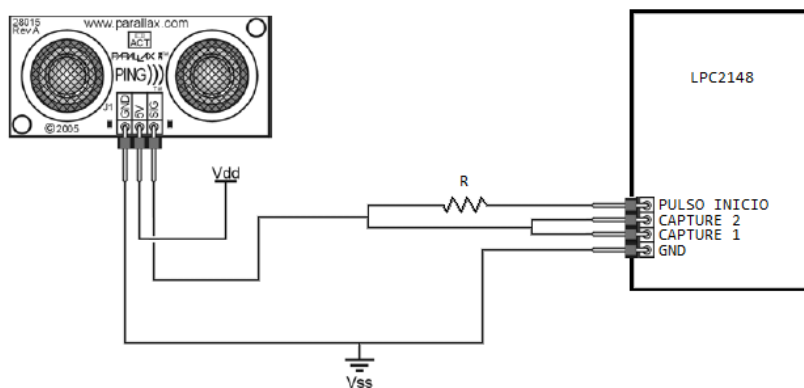
A continuación se detalla cada una:

- Vdd: Tensión de alimentación del sensor, según el datasheet, 5V.
- Vss: tierra.
- I/O: Pin de entrada salida. Es por este pin por el que se va a controlar el sensor, va a ser por el que se active y por el que se lea la información.

Para controlar este sensor, el procedimiento que se ha usado es el de usar tres pines del microcontrolador. Uno es el encargado de dar el pulso inicial de 5ms, esto se hace poniendo el pin a nivel lógico 1 durante 5ms.

Para llevar a cabo la medición del pulso de salida. Se pone en marcha un contador, y se habilitan dos pines denominados “CAPTURE”.

Uno de ellos, está preparado para leer un flanco de subida (inicio del pulso de respuesta), y el segundo para leer un flanco de bajad (fin del pulso de respuesta). Cada vez que se registraron estos flancos se almacenó el tiempo que llevaba transcurrido el contador. Una simple resta del tiempo de bajada con el de subida, nos dirá el tiempo de pulso de respuesta que nos ha dado el sensor. La conexión que se ha llevado a cabo es la siguiente:



**Figura 82:** conexión del radar

#### 1.8.3.4.3.2.2.2. Funcionamiento

El funcionamiento del sensor es sencillo pero hay que entenderlo.

No estamos ante un sensor de salida continua que podemos en cualquier momento. Si no que cuando deseemos realizar una lectura, debemos activarlo antes. Esto por qué se hace. El sensor debe de emitir un ultrasonido, el cual rebotará y será leído por el sensor. En base al tiempo que tarde el eco en llegar así será la respuesta.



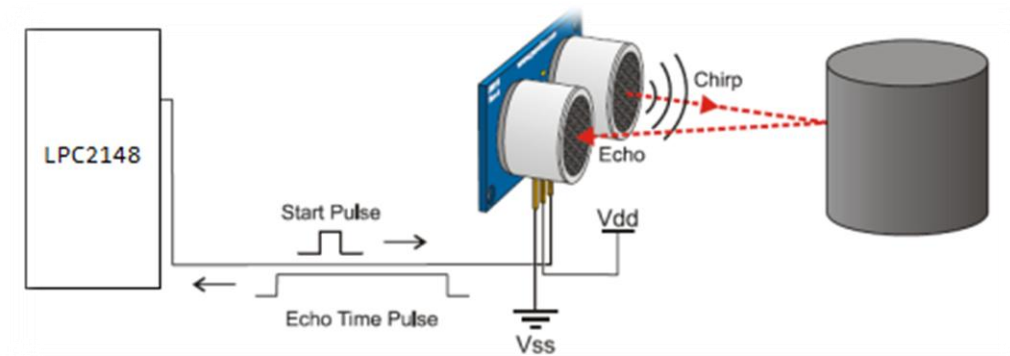
Pues bien, hay que activar el sensor y a continuación realizar la lectura. Los pasos a seguir son los siguientes:

- 1° Se le da un pulso en alto, de 5ms por el pin de señal.
- Se espera un delay especificado por el datasheet.
- A continuación se lee el pulso de respuesta.

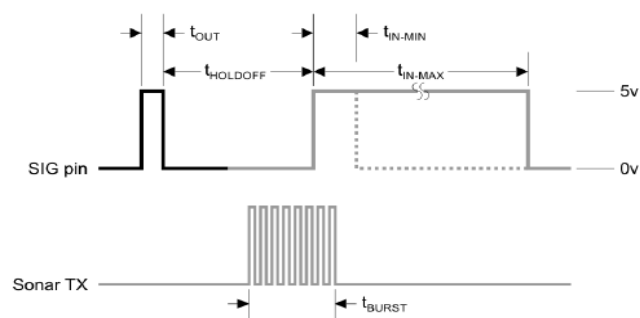
El sensor indicará la distancia a la que se encuentre el objeto sobre el que incidió el ultrasonido y devolvió el eco mediante un pulso de duración variable.

Contra más tarde el eco en regresar más lejos estará el objeto y más durará el pulso de salida. Este pulso ha de leerse con el microcontrolador.

En los siguientes esquemas se detallan el funcionamiento y duración de pulsos:



**Figura 83:** esquema funcionamiento radar

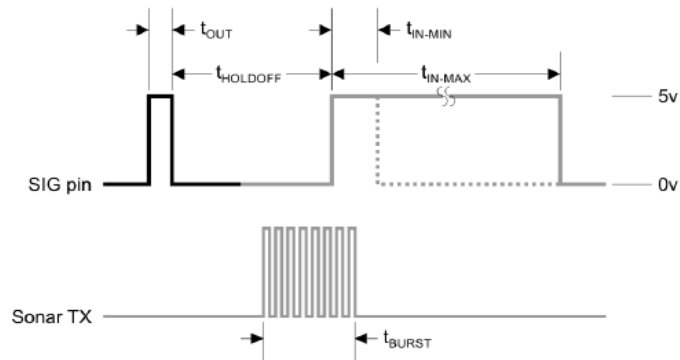


Host Device	Input Trigger Pulse	$t_{OUT}$	2 $\mu$ s (min), 5 $\mu$ s typical
PING)))	Echo Holdoff	$t_{HOLDOFF}$	750 $\mu$ s
Sensor	Burst Frequency	$t_{BURST}$	200 $\mu$ s @ 40 kHz
	Echo Return Pulse Minimum	$t_{IN-MIN}$	115 $\mu$ s
	Echo Return Pulse Maximum	$t_{IN-MAX}$	18.5 ms
	Delay before next measurement		200 $\mu$ s

**Figura 84:** tiempos de pulsos y respuestas en el radar

### 1.8.3.4.3.2.2.3. Salida

La salida que da el sensor será en tiempo, esta habrá que transformarla a distancia. Sabiendo que el tiempo es proporcional a la distancia, sabiendo el mínimo y el máximo se podrá interpolar un valor. En la siguiente tabla se dan todos los tiempos para trabajar con el sensor y los tiempos de respuesta:



Host Device	Input Trigger Pulse	$t_{OUT}$	2 $\mu$ s (min), 5 $\mu$ s typical
PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 $\mu$ s
	Burst Frequency	$t_{BURST}$	200 $\mu$ s @ 40 kHz
	Echo Return Pulse Minimum	$t_{IN-MIN}$	115 $\mu$ s
	Echo Return Pulse Maximum	$t_{IN-MAX}$	18.5 ms
	Delay before next measurement		200 $\mu$ s

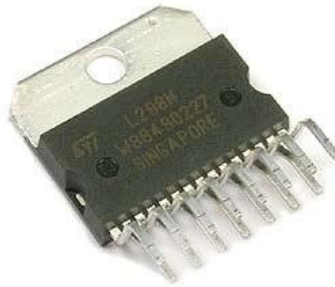
**Figura 85:** tiempos del radar

Hay que tener en cuenta el tiempo entre lecturas, no se puede realizar una nueva lectura si aún no ha pasado el tiempo mínimo para realizarla, en este caso 200us.

### 1.8.3.4.3.3. Módulo L298 y el motor

#### 1.8.3.4.3.3.1. Módulo L298

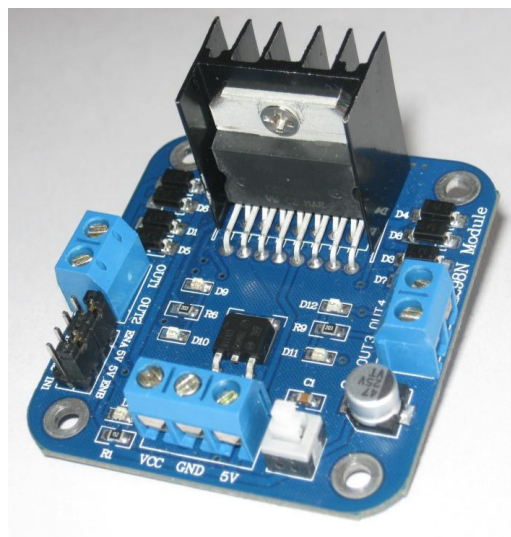
Para controlar el motor se ha usado un integrado que soportase una gran cantidad de corriente, debido al tamaño del motor. Este es el integrado L298N.



**Figura 86:** integrado L298

Este chip permite el control simultáneo de dos motores de corriente continua, en nuestro caso solamente controlaremos uno, el motor de tracción del vehículo.

Para usar este chip, se adquirió un módulo, (una placa ya diseñada y fabricada free-hardware comercializada en internet) que viene lista en la forma “plug & play”, solamente hay que conectar los pines, alimentación y motores, encenderla y funciona, la placa es la siguiente:



**Figura 87:** módulo L298

### 1.8.3.4.3.3.1.1. Funcionamiento

El esquema interno del L298 es el siguiente:

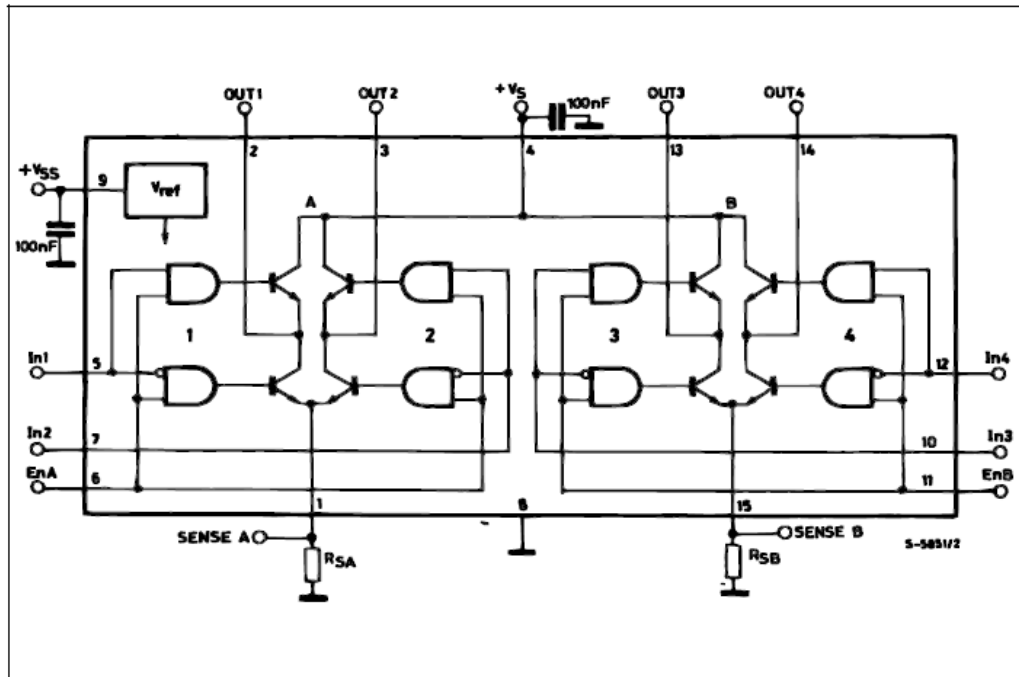


Figura 88: esquema interno L298

Como puede observarse el integrado consiste en un doble puente en H. Cada uno sirve para controlar un motor. Las entradas de que dispone son las siguientes:

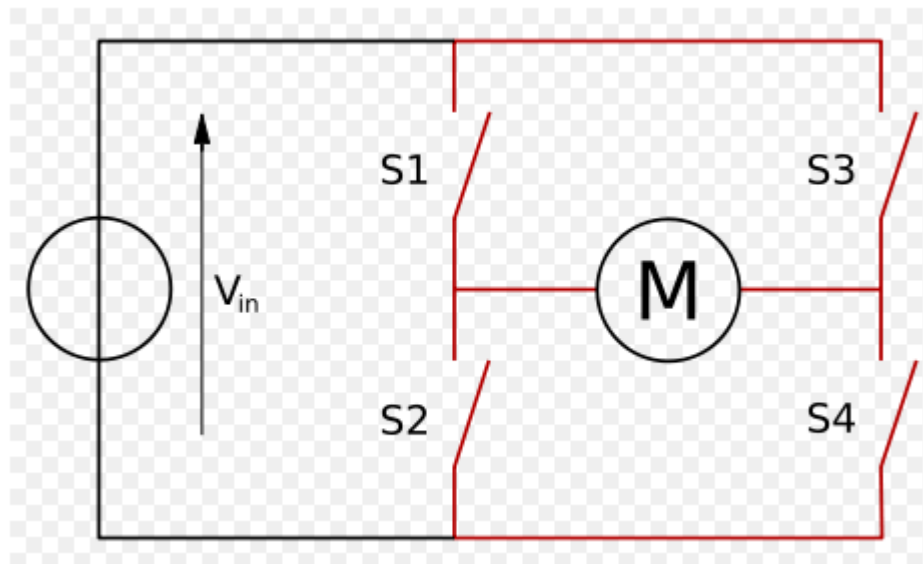
- In1, In2, In3, In4: Pines de control de cada puente, 1 y 2 para el puente A y 3 y 4 para el puente B.
- EnableA y EnableB: pines de habilitación de los puentes.
- Vs: tensión de alimentación del motor.
- Vss: alimentación del integrado.
- GND: masa
- Sense A y Sense B: Control de la intensidad que circula por el motor, en caso de no usarse han de estar conectados a masa, ya que si no el circuito del motor no se cierra y el motor no funcionaría.
- Out1 y Out2: pines de conexión para el motor.

Vemos que hay 4 puertas lógicas AND por puente, cada una activa un transistor del puente en el que esté. Los pines In1 e In2, son los que nos van a ayudar a controlarlas. Antes de nada. Saber que estas puertas lógicas están conectadas al pin Enable, que si no está habilitado a 1, ninguna funcionará. Por lo que Enable ha de estar habilitado para que el puente funcione.

Bien, In1 está conectado a dos puertas lógicas en diagonal, lo mismo ocurre en In2.

Cuando In1 esté habilitado e In2 no, la corriente circulará en un sentido por el motor y cuando esté al contrario lo mismo ocurrirá con la corriente.

Supongamos que ponemos nombre a cada conjunto puerta lógica – transistor del puente, los llamamos s1, s2, s3 y s4. Se van a comportar como interruptores de la siguiente forma:



**Figura 89:** esquema puente en H

Aquí, s1 y s4 están ligados al comportamiento de In1, y s2 y s3 al comportamiento de In2, vemos claramente que cuando se activa In1 la corriente circula en un sentido y cuando es In2 va en el contrario.

Cuando ambos están habilitados, ambos terminales del motor están al mismo potencial, lo que significa la clavada del motor y cuando ambos estén a cero, pues el puente estará desactivado por completo.

Siguiendo esta secuencia. Para controlar la velocidad del motor se ha utilizado la modulación por ancho de pulso, PWM (Pulse Width Modulator).

Si en vez de poner In1 por ejemplo, constantemente a 1, lo conmutamos, podremos variar la tensión media que llega al motor y así variar su velocidad.

Como se ha hecho esto, pues de la siguiente forma:

Hemos dicho que tenemos dos pines de control, In1 e In2, uno va a ser para la velocidad, el otro va a ser para el sentido de giro.

Por ejemplo, tomemos In1 para la velocidad, e In2 para el sentido de giro. Aunque el comportamiento de uno afecta al comportamiento del otro, a continuación se explica.

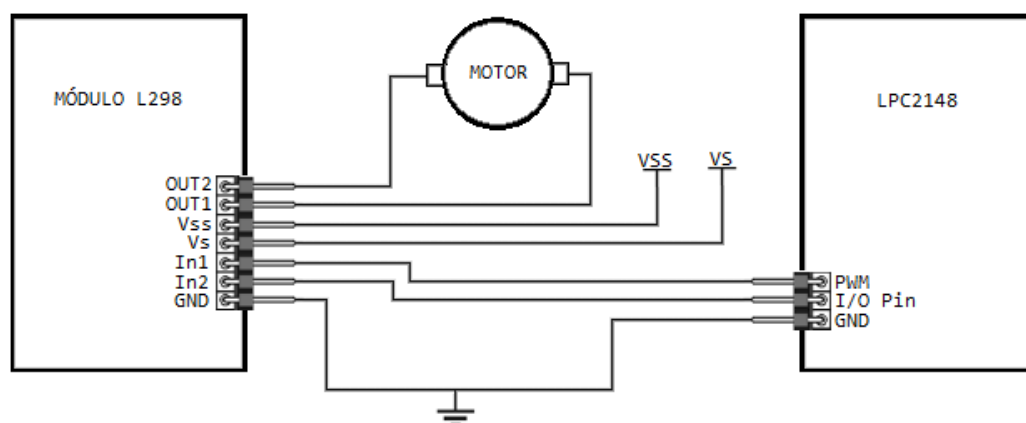
Si ponemos In2 a cero e In1 a 1, el motor girará en un sentido constantemente.

Si aplicamos ahora PWM a la señal In1, variaremos la velocidad del motor, ya que el motor girará cuando In1 esté activo. Entonces, con In2 a cero y PWM en In1 el motor girará en un sentido y la velocidad dependerá del PWM que se aplique.

Si deseásemos cambiar el sentido de giro, procederíamos de la siguiente forma. Hemos dicho que In2 controla el sentido, pues lo pondremos a 1. Que ocurre, si In1 está deshabilitado, el motor girará en sentido contrario, si In1 se habilita el motor se clava, y cuando In1 se deshabilite, pues girará. Este es el principio a seguir. En este caso, el PWM que aplicaremos en In1, debe de ser un PWM inverso, si deseamos que sea del 80%, debemos dar una señal del 20%.

### 1.8.3.4.3.3.1.2. Conexión

Se ha explicado anteriormente los pines de los que disponía el integrado para explicar su funcionamiento, a continuación únicamente se detalla su conexionado.



**Figura 90:** conexión módulo L298



### 1.8.3.4.3.3.2. Motor

Dado que el vehículo se ha construido a partir de un coche radiocontrol, se ha utilizado el motor que traía. El modelo es RS\_360SH



**Figura 91:** Motor RS360-H

Los parámetros del motor son los siguientes:

MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
	OPERATING RANGE	NOMINAL	SPEED r/min	CURRENT A	SPEED r/min	CURRENT A	TORQUE		OUTPUT	TORQUE		CURRENT
							mN-m	g-cm	W	mN-m	g-cm	A
RS-360SH-2885	3-9	7.2V CONSTANT	12500	0.36	10380	1.76	7.00	71.3	7.59	41.2	420	8.60
RS-360SH-10500	12-25	24V CONSTANT	7000	0.050	5600	0.20	4.50	45.9	2.63	22.5	229	0.80

**Figura 92:** especificaciones del motor

### 1.8.3.4.3.4. Servomotores

El vehículo dispone de tres servomotores, uno es para la dirección y los otros dos para la articulación. El servomotor elegido es el Hitec-HS311.

Se trata de un servomotor estándar el cual ya he usado anteriormente y va ideal con esta aplicación. Cada servomotor tiene un recorrido de 180° y una fuerza de 3kg/cm.

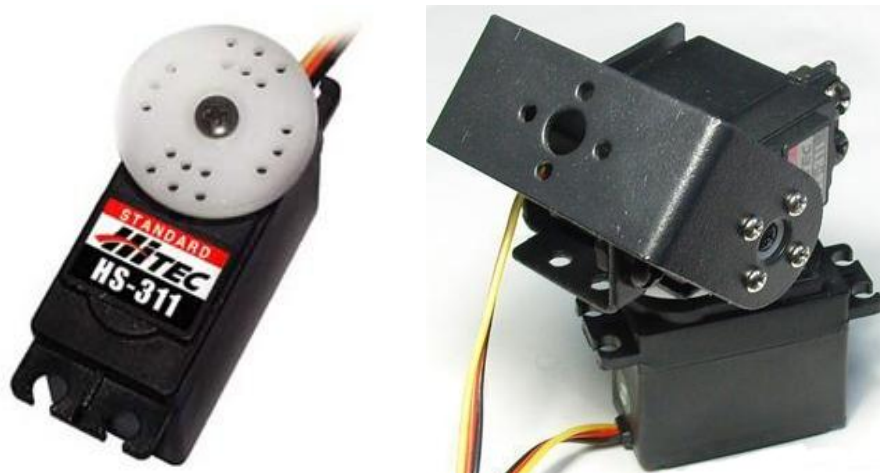


Figura 93: servomotor Hitec HS311 y articulación

#### 1.8.3.4.3.4.1. Funcionamiento

El funcionamiento de un servomotor es bastante sencillo. Se controla únicamente mediante un único pin.

Hay que transmitirle una señal, la cual consiste en una secuencia de pulsos, en la que se le da un pulso en alto que va de los 600us hasta los 1400us, seguido de un pulso en bajo de 20ms de duración. El valor que tenga el pulso en alto de 600 a 1400 será el ángulo de 0 a 180°.

El datasheet indica una tensión de alimentación de 4,5 a 6 Voltios. Contra más tensión se le da, más fuerza tendrá nuestro servo. Se aplicarán 5 Voltios. Con esto tendrá de fuerza de sobra para girar las ruedas de dirección y mover toda la articulación, levantando todo el peso que conlleva.

El uso de servomotores es la mejor opción ya que son posicionables, tienen una gran fuerza y se conoce en todo momento la posición en la que está.



### 1.8.3.4.3.5.1. Funcionamiento

Los láseres se activan con un transistor, el cual está conectado a un pin de entrada/salida del microcontrolador. La simple puesta a 1 del pin, polarizará al transistor en directa y este hará de interruptor. Cuando el pin se ponga a cero los láseres se apagarán.

El transistor utilizado es el BC547, se ha utilizado este porque soporta perfectamente la intensidad al utilizar ambos sensores. Además de tener lógica positiva, es decir, se polariza en directa cuando se le da un uno lógico y se pone en corte con un cero.

La tensión que se le da, es la recomendada por el datasheet, 5 voltios.

### 1.8.3.4.3.5.2. Conexión

La conexión realizada es la siguiente:

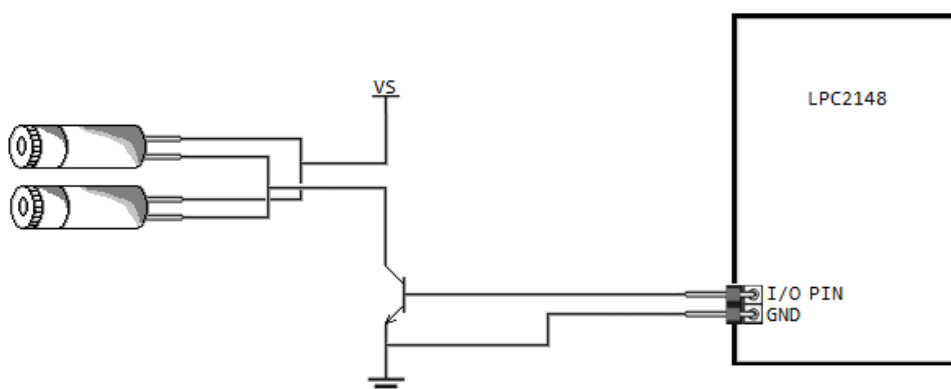
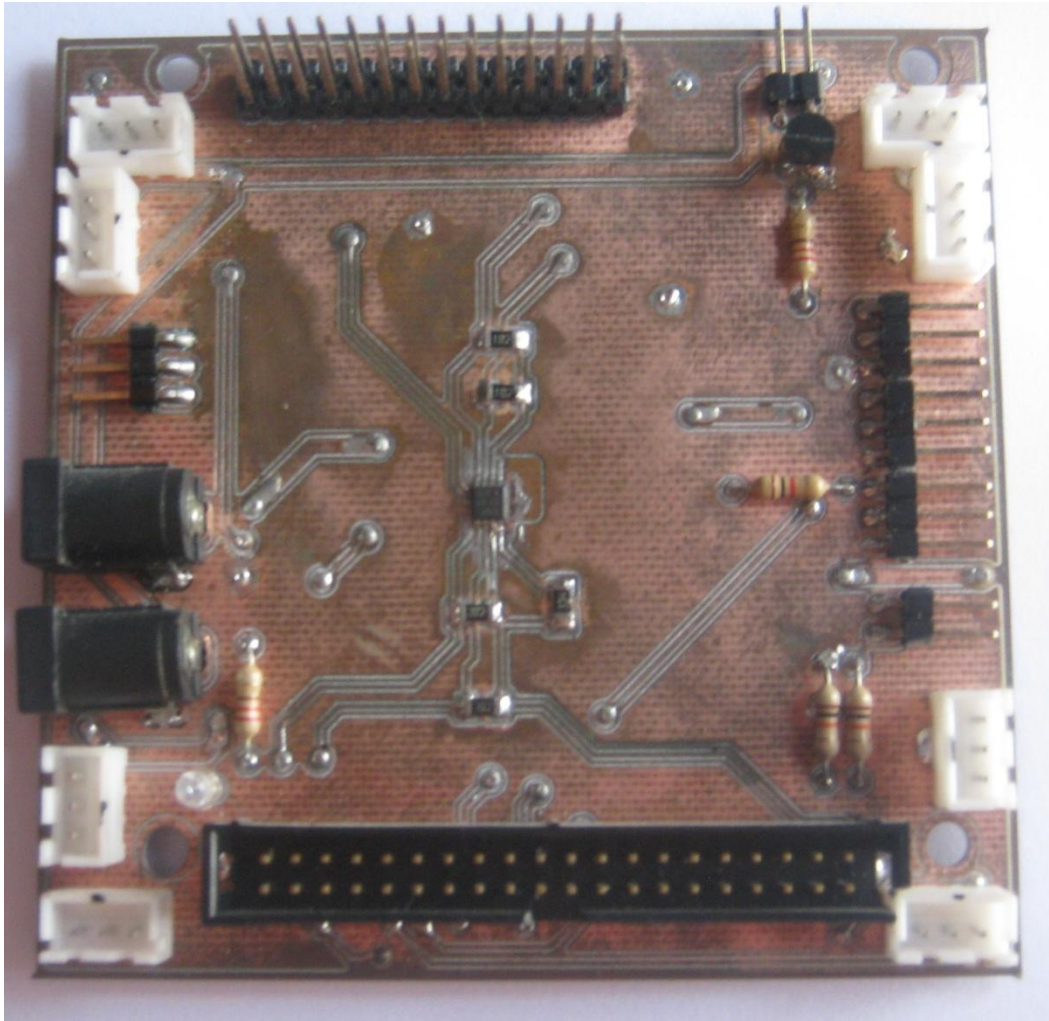


Figura 96: conexión de los láseres

### 1.8.3.4.3.6. Placa de enlace

El sistema consta también de una placa denominada de enlace, la función de esta placa es la de conectar "todo con todo". Hace de nexo de unión entre todos los elementos descritos anteriormente, el microcontrolador y la comunicación con el grupo dos.

Es en ella en donde se interconecta todo. La parte más importante de ella, es sin duda el conversor lógico de niveles para la comunicación i2c, se detalla más adelante.



**Figura 97:** placa de enlace

En esta placa están implementados todos los circuitos anteriores, para ver su esquemático ir al apartado esquemáticos.

Para la comunicación i2c entre el grupo 1 y el grupo 2, hace falta una conversión lógica de niveles, debido a que el i2c del microcontrolador funciona a 3,3 Voltios y el del grupo 2 a 1,8. Así que hubo que hacer un conversor.

Dicho se llevo a cabo usando el integrado PCA9306, un conversor de niveles lógicos para adaptar comunicaciones I2C que funcionan a diferentes tensiones. Perfectamente válido para esta aplicación. Primero se realizaron pruebas por separado una vez hecho el circuito para ver que transmitía de un lado a otro y posteriormente se llevo a la conexión directa en el vehículo.





### 1.8.3.4.3.7. Software global

El programa interno que lleva el micro funciona de la siguiente manera.

Es un “esclavo a la espera de órdenes”. La comunicación le entra por i2c, estando el micro configurado como esclavo. Cuando le lleguen los datos por el bus de comunicación, estos serán interpretados.

Dependiendo de los datos entrantes así serán la respuesta del microcontrolador.

Como se explicó anteriormente, los datos entrantes se corresponden con órdenes que llegan. Estas órdenes pueden ser de ejecución o de devolución.

El software está preparado para interpretar cualquier secuencia de datos entrante.

Aunque responda cuando se le da una instrucción. Si no hay datos entrantes por la comunicación. El micro sigue una secuencia constante.

De manera cíclica y continua, se leen todos los sensores que tiene conectados a él. Capta cada valor y lo almacena en un buffer. Denominado buffer de sensores. Cada valor se almacena en una posición determinada.

Cada nuevo valor de sensor se escribe encima del anterior, puesto que un valor anterior no va a ser de efecto alguno, más aún, sería perjudicial, dado que no se correspondería la lectura con la realidad.

Cuando entren datos indicando una orden de devolución, diciendo que se quiere leer un determinado sensor, el software responderá, leyendo la posición del buffer destinada al sensor que se está pidiendo y enviará este dato por el bus de comunicación. Solo se hará una lectura por ordenanza. Es decir, que no se puede requerir más de un sensor a la vez.

El microcontrolador tiene programadas una serie de funciones para el control de los diferentes dispositivos conectados a él, tales como son sensores, motores etcétera.

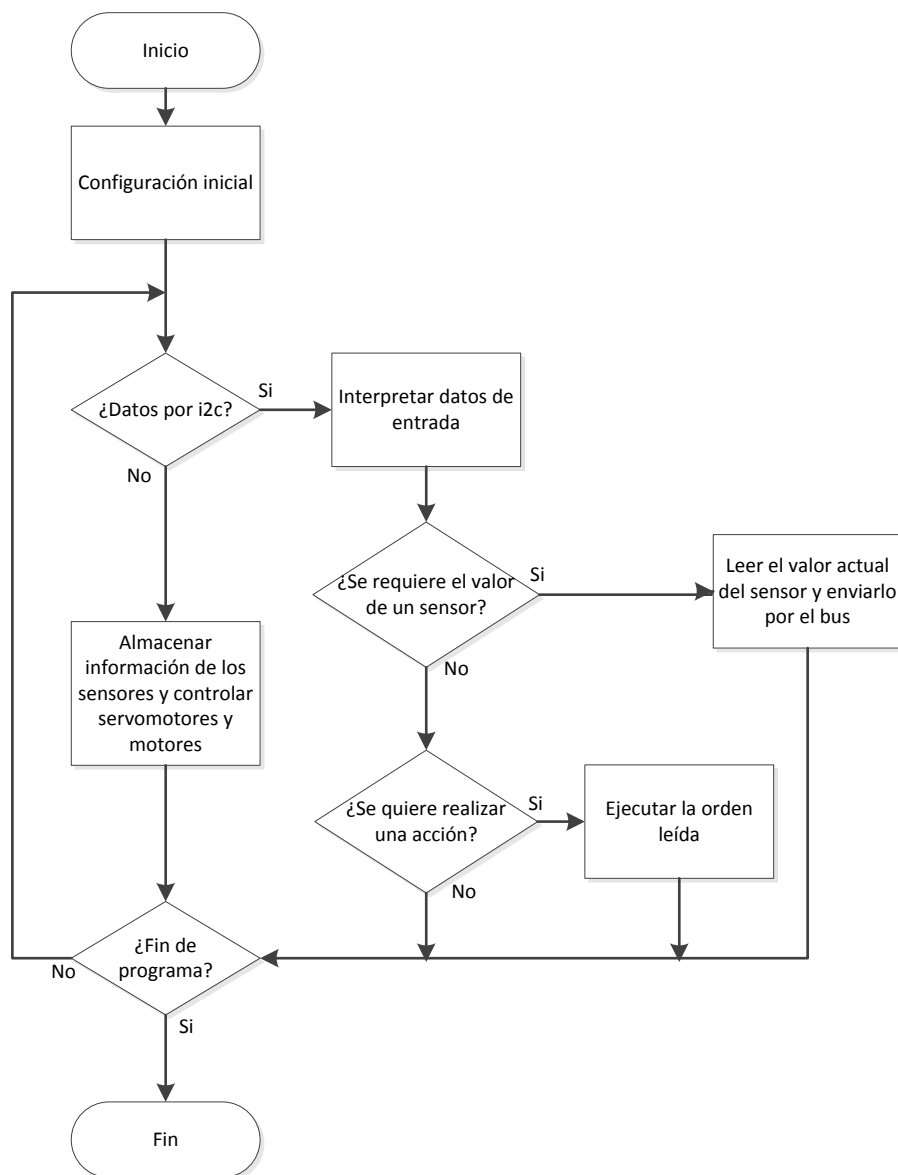
Si la orden entrante es de ejecución. Por ejemplo, posicionar un servomotor a 30°, pues los datos entrantes contendrán un código relacionado con el servomotor, con posicionarlo y el valor de 30

Cada orden que entre será ejecutada sin dilación, sin duda, cumpliendo así las leyes del sistema. En la que ningún elemento situado en un puesto inferior en la escala del sistema puede dudar sobre una orden venida de un nivel superior.

A continuación se detalla cómo funciona el software para trabajar sobre cada uno de los elementos que tiene a su disposición. Sensores, motores etc, etc.

Cada uno tiene su clase propia, en donde se concentran todas las funciones, variables y operaciones escritas para cada uno. A partir de la clase se ha creado cada objeto propio de cada elemento, pudiendo mantenerse así una hermeticidad respecto a cada uno, con la que ningún procedimiento de una clase puede interferir en la de otro, causando así un mal funcionamiento.

A continuación se muestra el diagrama de flujo del programa en general. Es bastante sencillo, aunque no es totalmente fiel a la realidad, pero se entiende mejor. En él se ha mostrado un bucle, que lee constantemente si hay datos que llegan por i2c, esto realmente ocurre con una interrupción. Puede ocurrir en cualquier momento, pero a fines prácticos, es igual el diagrama y se entiende más claramente.



**Figura 100:** diagrama de flujo del microcontrolador

### 1.8.3.4.3.8. Software individual

#### 1.8.3.4.3.8.1. Motor y Servomotores

Tal y como se ha explicado anteriormente cada uno de ellos se controlan mediante PWM. El problema es que el modelo de microcontrolador utilizado solo dispone de un solo generador de PWM. Este es un módulo ya preparado en el microcontrolador, que funciona independiente del resto del micro. Lo que es una gran ventaja, dado que se puede poner a funcionar y hacer otras cosas al mismo tiempo.

Dado que solo hay se podría usar el mismo para todo, pero la base de tiempo no es la misma para un servomotor que para un motor. Se ha optado por usar el generador de PWM para el motor y generar PWM ficticio para los servomotores.

A continuación se explica cómo se ha procedido para trabajar con cada uno.

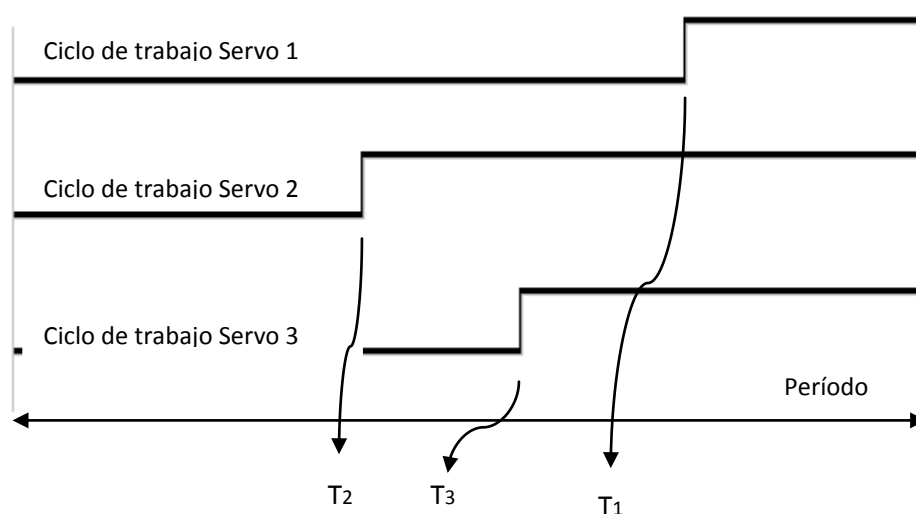
##### 1.8.3.4.3.8.1.1. Servomotores

Se va a generar PWM ficticio para los tres servomotores de que consta el sistema. Esto se hará mediante interrupciones. Para ello, se dispondrán tres pines de entrada y salida, uno irá a cada servomotor. Se utilizará un timer para los períodos.

Una vez establecida la base de tiempo, habrá una variable de tiempo para cada servo. Que controlará la frecuencia de cada PWM generado. Esto lo haremos mediante interrupciones.

Al inicio de ciclo cada pin se pondrá a cero. A medida que el temporizador vaya alcanzando cada tiempo, se irán poniendo los pines a uno. Hasta que termine el periodo que se pondrán todos a cero.

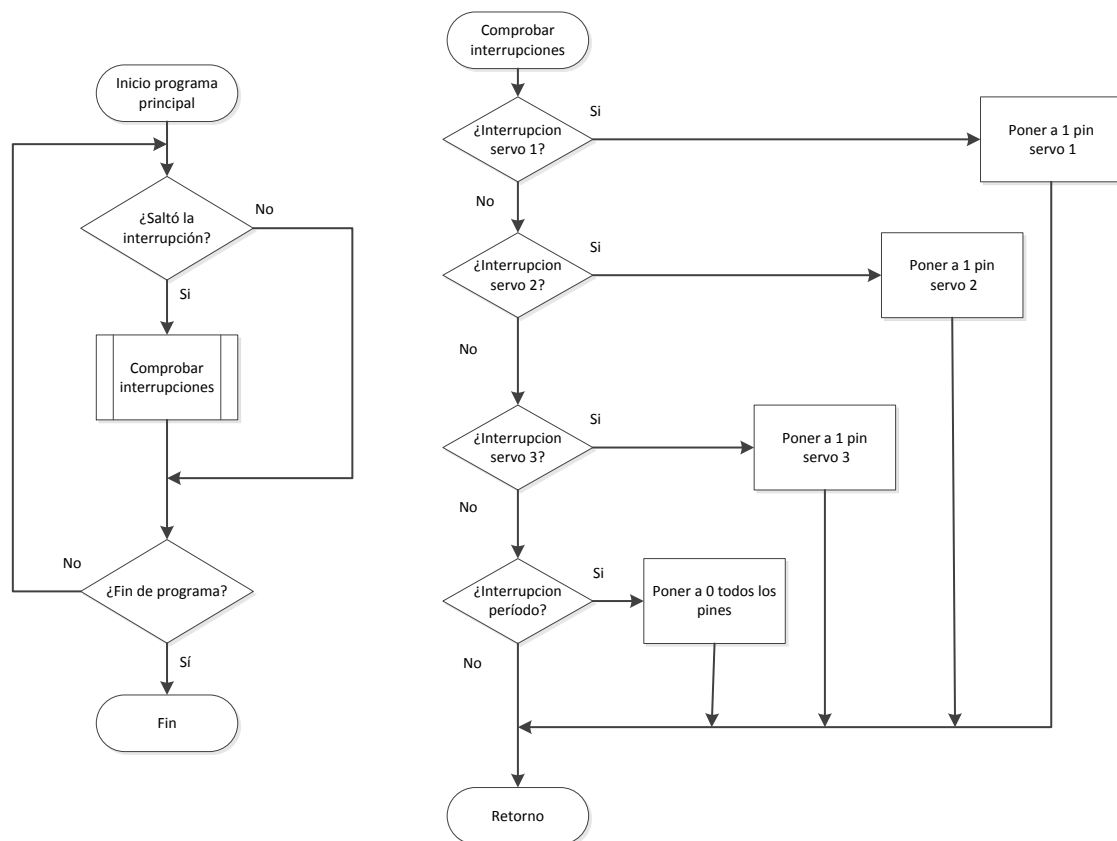
En el siguiente esquema se aprecia mejor:



**Figura 101:** funcionamiento del pwm para los

T1, T2 y T3, serán los tiempos en los que saltaran las interrupciones y el pin que corresponda se pondrá a nivel lógico 1. Cuando acabe el período, los tres se pondrán a nivel lógico 0. Así continuamente se manejarán los servomotores. Para variar el ciclo de trabajo solamente habrá que actuar sobre estos tiempos, al variarlos, variará cuando se ponen a uno los pines.

A continuación se muestran los diagramas de flujo de cómo funciona la generación de PWM para el control de los servomotores. El de la izquierda muestra el funcionamiento en sí, y el de la derecha muestra a detalle la comprobación de las interrupciones.



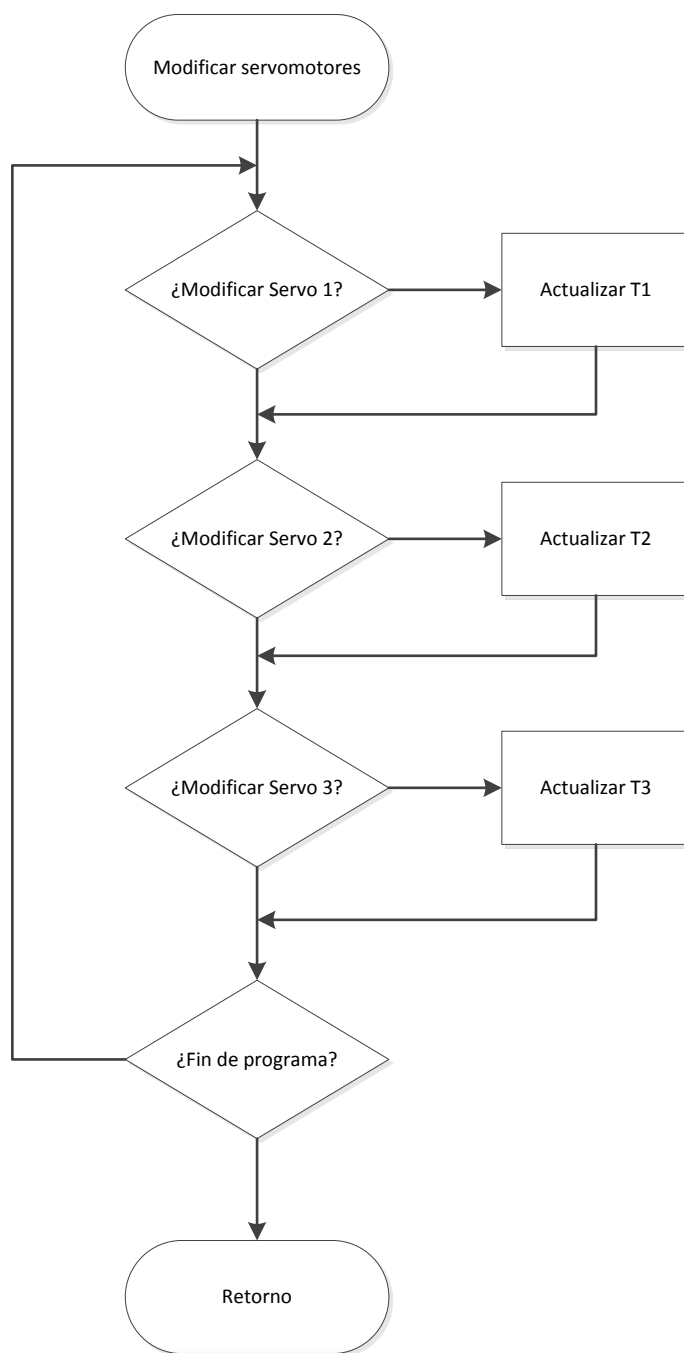
**Figura 102:** diagrama de flujo de las interrupciones de los servomotores

A continuación se incluye como funcionará el programa de las funciones que modifican los valores de los servomotores.

Cada vez que se quiera mover un servomotor, la función admitirá como argumento el valor en grados del ángulo en el que se quiere que se coloque, sabiendo que el servo se posiciona de 0° a 180°.

Se realizará una conversión, de grados a tiempo, siendo conocidos los intervalos de tiempo en los que funciona el servo.

Así se crearon estas funciones, una vez hecha la conversión a tiempo, al variable T2 se actualizará con el valor que haya resultado del proceso y provocará que el servo se posicione.



**Figura 103:** diagrama de flujo de la llamada a modificar los servomotores

Como se observa es bastante sencillo, solamente se ha creado una función, que es la de posicionar un servomotor, esta se aplica por igual a todos los servomotores dados que todos son del mismo fabricante con las mismas características. Si hubiese habido uno de los servos distintos habría que haber hecho una generación de PWM distinto y con lo cual funciones distintas.

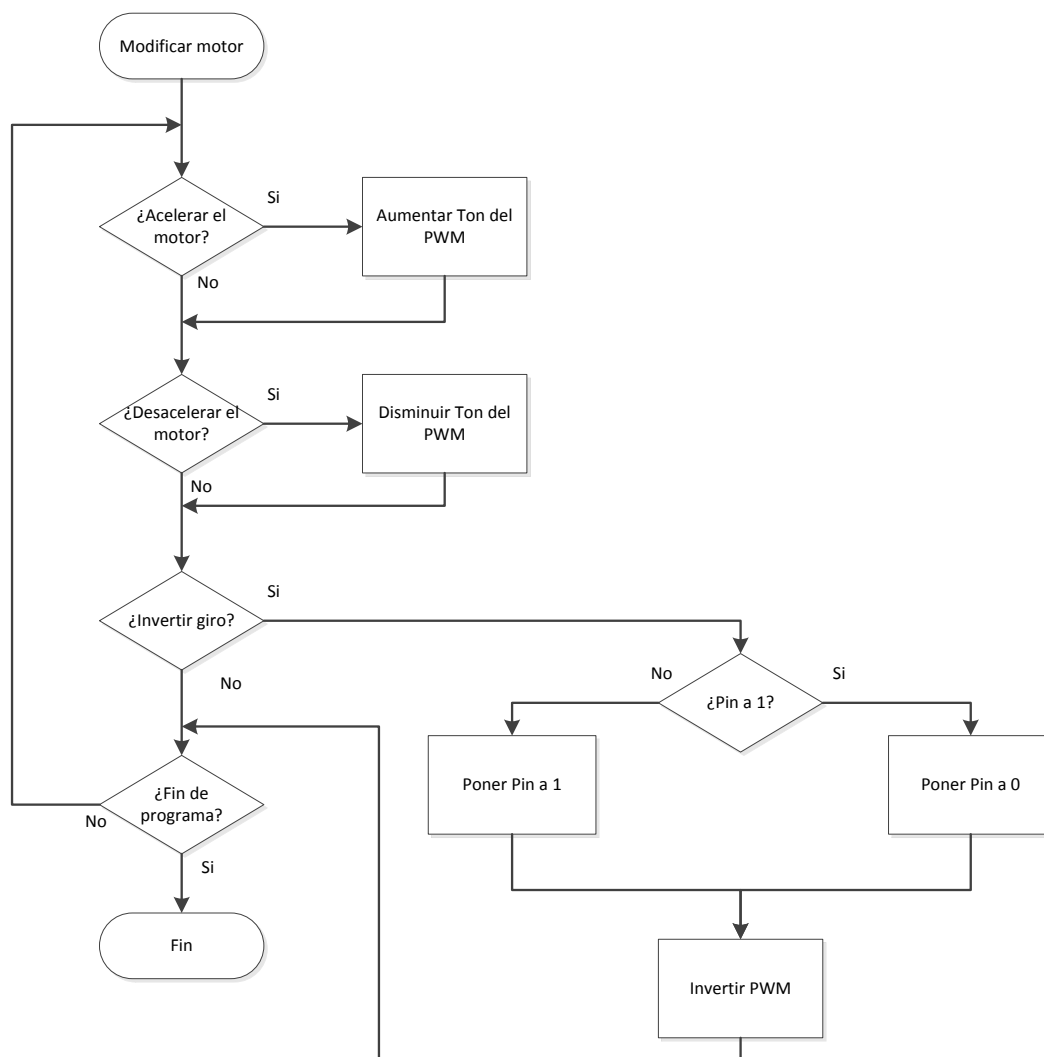
### 1.8.3.4.3.8.1.2. Motor

El control del motor se ha explicado en su correspondiente sección anteriormente. Para usarlo hemos utilizado el generador de PWM que incorpora consigo el microcontrolador. Su control es realmente fácil, dado que no hay que usar interrupciones, solamente hay que indicarle el período y el tiempo en alto. El mismo se encarga de generar la señal a través de un pin, que habrá que configurar para que se conecte al generador de PWM

A parte del generador de PWM, necesitaremos de un pin de entrada y salida. Con estos dos elementos ya tendremos lo necesario para controlar el motor.

Únicamente tendremos que controlar el tiempo en alto variando el registro del generador de PWM correspondiente y cambiar valor del pin de 1 a 0 o viceversa.

Se muestra el esquema de cómo funcionará el modificar el motor:



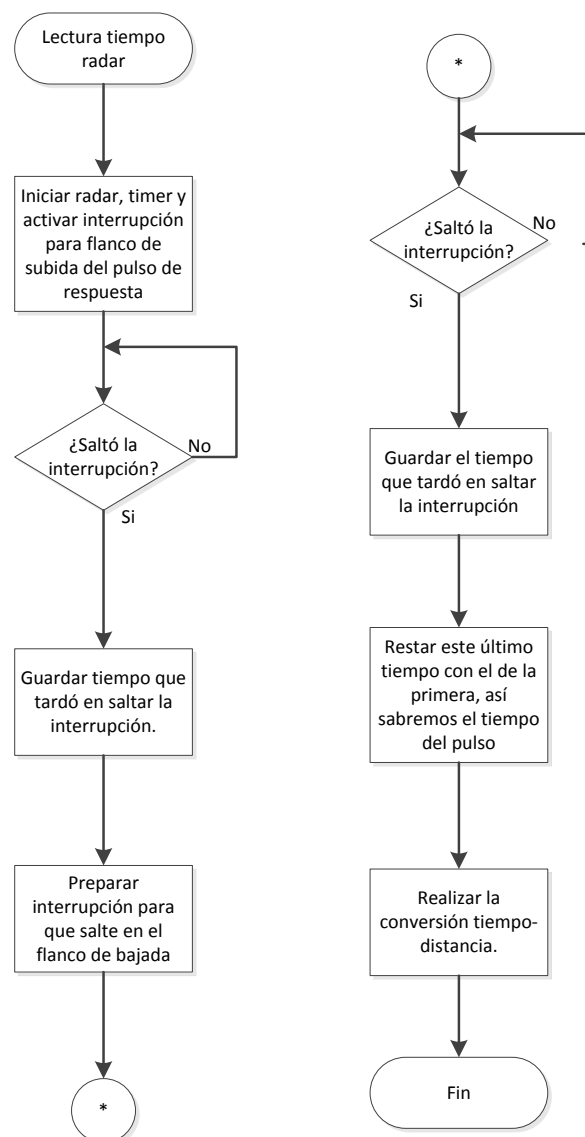
**Figura 104:** diagrama de flujo del motor



### 1.8.3.4.3.8.2. Radar

El funcionamiento del radar se ha explicado anteriormente, pero programarlo no fue tan fácil, por el siguiente motivo. Cuando se le da el pulso de inicio, para que empiece la medición, el pulso de respuesta máximo puede ser de 18500 ms. Lo que es una gran cantidad de tiempo para dejar parado todo el sistema a la espera únicamente de medir el pulso de entrada. Para solucionar esto, y no perder tiempo, se han recurrido también a las interrupciones. Cuando se inicia el radar, se activa la interrupción. Ahora se continua con la rutina normal del microcontrolador.

Cuando llega el pulso que devuelve el radar, salta una interrupción y se guarda el momento en el que llegó. Se vuelve a salir y cuando termine el pulso, salta otra interrupción, se guarda el tiempo y se le resta el anterior, así se obtendrá la medida sin perder tiempo en una espera demasiado larga.



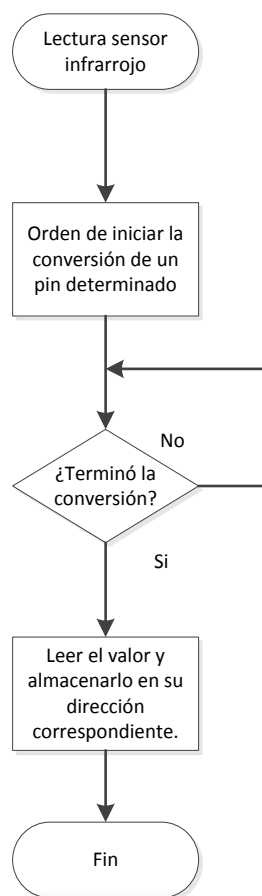
**Figura 105:** diagrama de flujo para leer el radar

### 1.8.3.4.3.8.3. Sensores infrarrojos

La programación de los sensores es bastante sencilla, dado que no hay interrupción alguna, si no que se ha escrito una función que lo que realiza es una conversión analógica-digital del valor del sensor.

Para leerlos el microcontrolador trae incorporados dos convertidores analógicos-digitales, con una serie de pines conectados a él, que habrá que configurar.

Cuando se precise, se le encargará la conversión y se podrá leer el valor resultante en un registro determinado. Así se efectuará con todos.



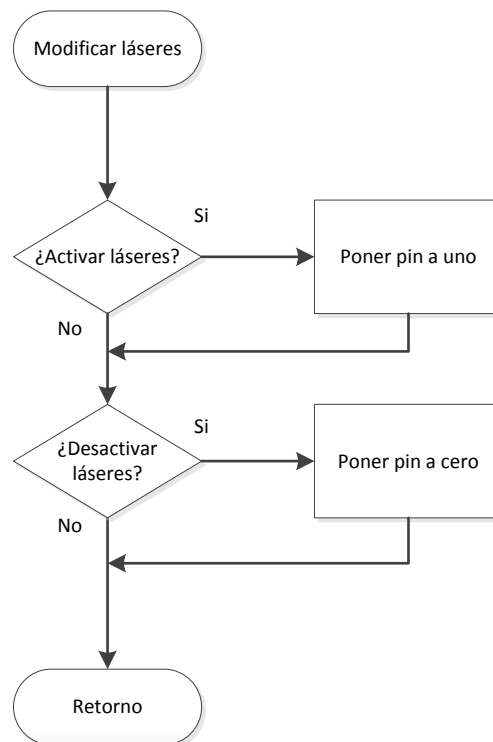
**Figura 106:** diagrama de flujo para leer un sensor infrarrojo

Como se observa esta se podrá efectuar con cualquier sensor infrarrojo. Es cierto que hay un tiempo de espera entre lectura y lectura del sensor, pero debido a que se están leyendo los ocho uno detrás de otro, el tiempo desde que se lee un sensor hasta que se vuelve a leer, es lo bastante grande para superar al tiempo mínimo entre lecturas de un mismo sensor.

Con lo cual no se ha precisado de un delay entre lectura y lectura.

### 1.8.3.4.3.8.4. Láseres

El funcionamiento de los láseres es bastante sencillo, simplemente hay un transistor conectado a un pin, cuando este se establezca a uno, los láseres se activarán y cuando se ponga a cero los láseres se desactivarán.



**Figura 107:** diagrama de flujo para trabajar con un láser

La única programación que lleva es la de configurar el pin como salida y crear una función que tome como argumento si se activan o no.

### 1.8.3.4.3.8.5. I2C

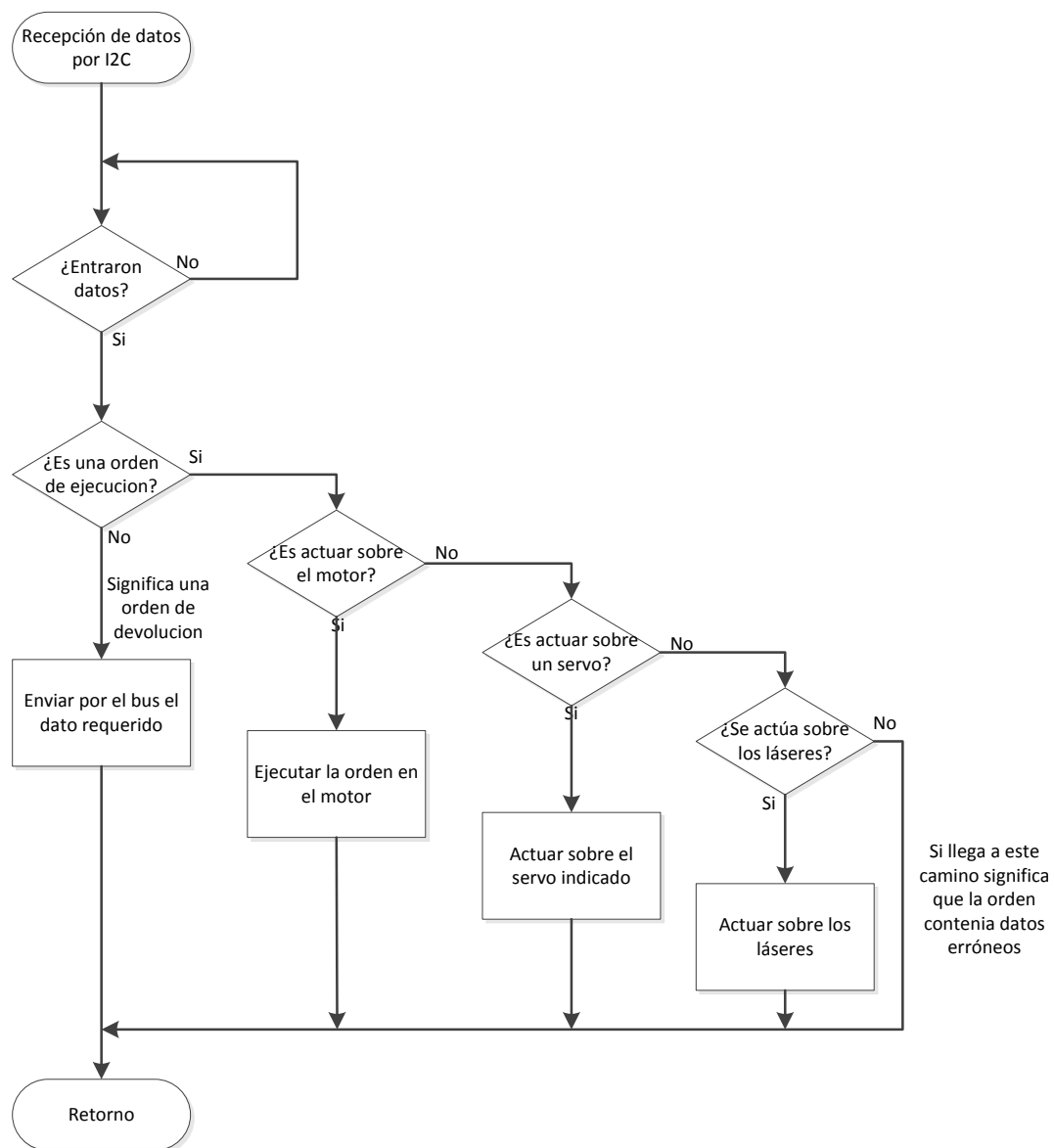
Esta es sin duda la programación más ardua que ha tenido el microcontrolador.

Antes de implementarla se ha requerido de un estudio previo a la programación, de todo el protocolo i2c y del manual del módulo i2c del microcontrolador. Este manual viene preparado para comunicarse pero lleva su funcionamiento propio.

Se partió de unas funciones proporcionadas por el profesor Víctor Manuel Sánchez Corbacho, a partir de dichas funciones, se fueron construyendo unas nuevas más acorde a nuestra programación en general.

El i2c se puede configurar como esclavo o como maestro, el microcontrolador está siempre configurado como esclavo a la espera de datos, una vez que entren datos, se comprobarán si son, como se dijo, ejecutar una orden o devolver un valor.

El esquema de su funcionamiento es el siguiente:



**Figura 108:** diagrama de flujo del funcionamiento del i2c

## 1.8.4. Alimentación

### 1.8.4.1. Batería

La mejor opción para alimentar el coche fue una batería de Lipo, de larga duración y sin efecto memoria.

El voltaje se decidió en base al que se necesitaba para el motor, es decir, la batería del coche antiguo era de 9,6 Voltios, pues se seleccionó una batería superior a esta, para asegurarnos el buen funcionamiento del motor.

La elección hecha fue una batería LiPo 3S.



**Figura 109:** batería LiPo 3S

Esta batería está compuesta por tres células de 3,7 Voltios cada una, colocadas en serie, obtienen una tensión en bornas de 11,1 Voltios.

Aunque dicen 11,1V a máxima carga se obtiene de cada célula a 4,2 Voltios, con lo que se pone la tensión a 12,6 Voltios.

Esta tensión es más que suficiente para suplir todas las diferentes tensiones que necesita el coche, ya que la máxima es el motor, que se alimentará directamente a esta tensión.

En base a la intensidad, vemos que la batería indica 5000mAh. Esto significa que es capaz de estar supliendo durante una hora, 5000 mA. Puesto que se estimó el consumo total del coche en 3000 mA, Se obtiene una autonomía de más de una hora de funcionamiento.

Si en algún momento, hay una demanda fuerte de potencia, y la corriente que tiene que suministrar la batería, supera los 5000 mA, la batería viene preparada para ello. Más aún, observamos que la batería 20C, esto dice que puede suministrar en 20 Veces, la intensidad nominal, es decir,  $20 \times 5000 = 100000 \text{mA}$ .

### 1.8.4.2. Fuente de alimentación

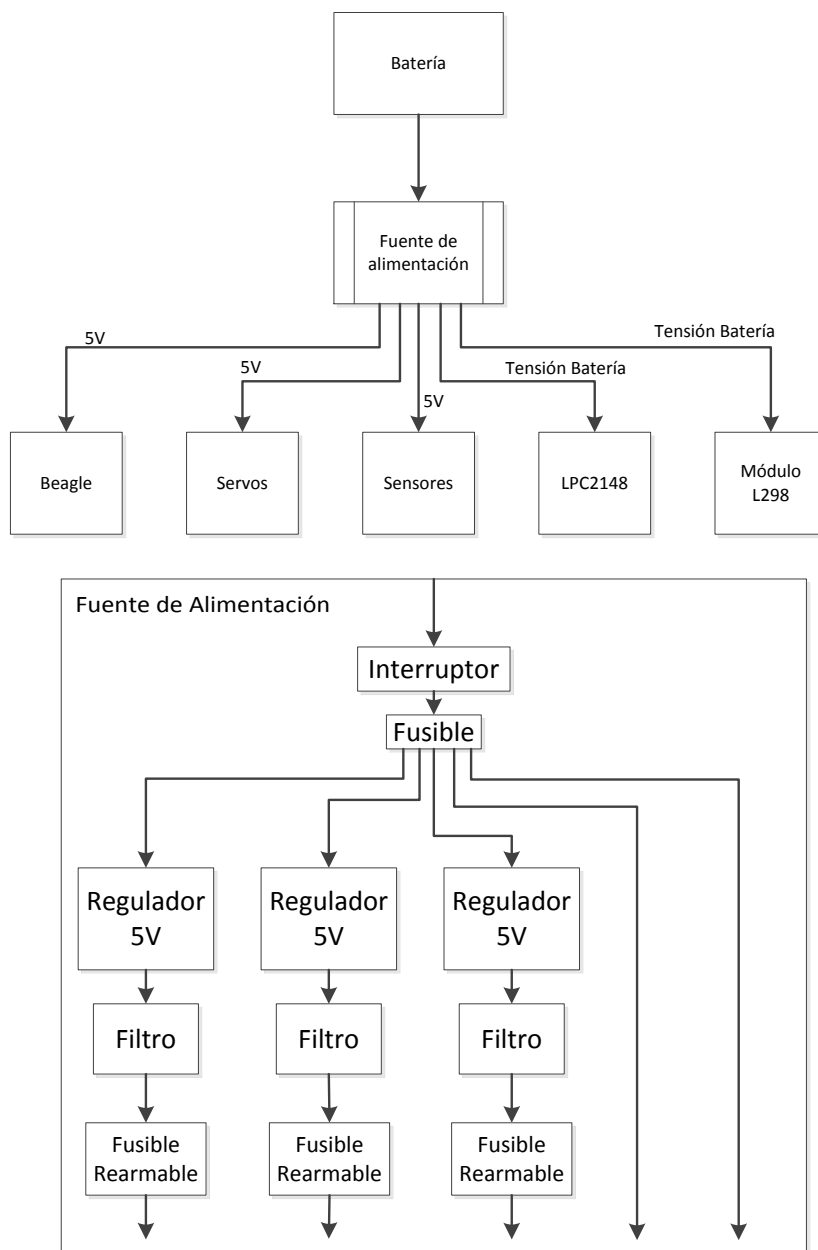
Esta batería se tiene que distribuir a todo el vehículo, para hacerlo se ha fabricado una fuente de alimentación, a la que se conecta la batería y salen todas las demás conexiones para los demás dispositivos, las tensiones que necesita cada dispositivo se reflejan en la siguiente tabla.

ELEMENTO	VOLTAJE	mA
SHARP0	5 V	35 mA
SHARP1	5 V	35 mA
SHARP2	5 V	35 mA
SHARP3	5 V	35 mA
SHARP4	5 V	35 mA
SHARP5	5 V	35 mA
SHARP6	5 V	35 mA
SHARP7	5 V	35 mA
RADAR	5 V	35 mA
Servo Dirección	5,5 V	200 mA
Servo Articulación Vert.	5,5 V	200 mA
Servo Articulación Horiz.	5,5 V	200 mA
Motor	Tensión de alimentación	600 mA
Módulo L298	Tensión de alimentación	40 mA
Laser 1	5 V	2mA
Laser 2	5 V	2mA
Adaptador Wi-Fi USB	5 V	200 mA (Estimación)
Cámara digital	3 V (Conectada a beagle)	50 mA
BeagleBoard	5 V	800 mA
LPC2148	5 V	150 mA
Suma Total		2759 mA
<b>Suma más Tolerancia 15%</b>		<b>3170 mA</b>

Figura 110: tabla de intensidades necesarias

Como se observa en la tabla, hay que hacer tres regulaciones a cinco voltios. Los otros dos elementos, se conectan directamente a la batería sin ningún tipo de operación de por medio debido a que ellos ya tienen su propio regulador de tensión y protector de corriente. La fuente de alimentación tiene en su interior un fusible rearmable por cada salida regulada, los elementos a los que no se les regula la tensión disponen ellos mismos de sus propias protecciones. Al lado de cada salida regulada, se ha implementado un indicador LED azul, para verificar visualmente que la salida está operativa.

A continuación se muestra el esquema de cómo funciona la regulación y la fuente de alimentación interiormente.



**Figura 111:** Funcionamiento fuente de alimentación



### 1.8.4.3. Reguladores de tensión

Para desarrollar esta fuente de alimentación, se pensó en utilizar primeramente los reguladores lineales 7805. Estos reguladores tienen una salida bastante estable.

El problema está en que al ser lineal, disipará una gran cantidad de calor, este calor debe de ser evacuado.

Debido al sitio en el que está la fuente de alimentación, evacuar el calor sería un gran problema.

Además, el calor disipado es energía que se está desperdiciando, energía que se pierde de la batería, con una consecuente bajada de autonomía.

Para suplir estos problemas se eligieron reguladores de tensión conmutados. Estos aparte de no disipar calor, no desperdiciarán energía.

El modelo elegido fue el OKI-78 SR de Murata.

Tienen un alto coste, pero se suple con la eficiencia de la fuente de alimentación. También tiene la característica que es capaz de suplir 1,5 Amperios. Lo que lo hace realmente potente.



**Figura 112:** regulador de tensión conmutado

## 1.9. Líneas futuras de desarrollo

- La versatilidad del proyecto ante el que estamos, permite que pueda seguir desarrollándose en múltiples campos.
- En mi caso, es un vehículo para vigilar. Pero no se quita la posibilidad de que se use para otra aplicación, por ejemplo la exploración en busca de algo concreto, la observación de un terreno, exploración de zonas catastróficas, exploración de zonas peligrosas.
- Se le pueden incorporar multitud de dispositivos. Se puede modificar la forma para incluirle por ejemplo brazos y una zona de carga, pudiendo así servir para recoger elementos determinados.
- Se puede seguir avanzando en la toma de decisiones, que cada vez más entrenado y le sea más fácil la tarea del movimiento por ejemplo. Se podría hacer que aprenda, que aprenda de los errores y de características mismas que el descubra.
- En base a la comunicación, este se ha hecho mediante una red Wi-Fi, pero se le podría incorporar una comunicación de tercera generación, y transmitirse con el usuario a través de la red de telefonía y datos de alguna compañía, con lo que para trabajar solo necesitaría que en la zona hubiese cobertura.
- Añadirle un GPS, con lo que podría saberse la posición exacta en la que estuviese, además de incluirle un mayor número de sensores para saber su colocación física exacta.
- Añadirle más cámaras, para poder tener una visión global en todo momento de lo que ocurre a su alrededor, dado que ahora mismo está limitado a una sola.
- Si se le recubre de un material protector, podría intervenir en situaciones de peligro de explosión, sin tener que poner en peligro personas humanas.
- Un punto para desarrollar también, sería crear varios prototipos y lograr que se comuniquen unos con los otros, que la información disponible en uno la pueda adquirir también otro.
- En general estamos ante un proyecto de una versatilidad tan grande como la imaginación de la persona que lo tenga, dado que simplemente es un vehículo móvil al que se le ha puesto una programación.
- Si se quiere que sirva para alguna cosa en concreto, basta con añadirle los dispositivos, si hicieran falta en su caso, cambiarle la programación y listo.

## 1.10. Planificación

ID	Nombre de la tarea	Duración	Octubre 2011	Noviembre 2011	Diciembre 2011	Enero 2012	Febrero 2012	Marzo 2012	Abril 2012	Mayo 2012	Junio 2012	Julio 2012	Agosto 2012	Septiembre 2012
			1	2	1	2	1	2	1	2	1	2	1	2
1.	Preparar coche antiguo													
2.	Diseño y fabricación de piezas de metacrilato													
3.	Diseño, fabricación y soldadura de placa de enlace y fuente de alimentación													
4.	Montaje estructura del vehículo													
5.	Instalar placas electrónicas													
6.	Instalar servomotores y sensores													
7.	Programar microcontrolador													
8.	Programación Beagle													
9.	Programar interfaz de usuario													
10.	Funciones de movimiento autónomo													
11.	Documentación													

## 1.11. Conclusiones

Durante todo este tiempo no he dejado de aprender, cada día, cada vez que hablaba con alguien o cada vez que intentaba hacer algo nuevo, he llegado a comprender bien el significado de la frase “Cuanto más sé, más me doy cuenta de que menos sé”.

Esta experiencia ha sido grandiosa, he profundizado conocimientos en electrónica, programación, microcontroladores, sistemas embebidos, procesamiento de imágenes, autonomía de robots, robótica en general, uso de maquinaria, herramientas, diseño, montaje y una gran lista de más cosas.

Ha sido un antes y un después en la carrera, aquí no trabajas para aprobar, se trabaja para superarse a uno mismo. Para cosas que antes solo se conocían de teoría verlas aplicadas realmente en algo.

Que el hecho de que te digan “Eso es muy difícil”, “No vas a poder”, “Es mucha tela”, “¿Por qué no haces algo y te lo quitas de en medio ya?”, solo sirve para motivarte aún más y querer llegar siempre más lejos.

He podido ver como muchos alumnos, toman el proyecto fin de carrera como si de un mero trámite se tratase. Gracias al trabajo realizado, a la gente que he conocido, todo lo aprendido, tomo al proyecto fin de carrera, como una oportunidad de poder hacer algo creado y pensado por uno mismo, teniendo el apoyo de profesores, materiales disponibles en la escuela y alumnos que te ayudan.

No hay duda de que en el futuro deseo seguir aprendiendo más aún en el tema, ya sea trabajando para alguna empresa o por cuenta propia para disfrute personal.

Sobre todo, la mayor conclusión que se saca, es que, contra más te divierta algo, menos te costará realizarlo.



## 2. Anexos



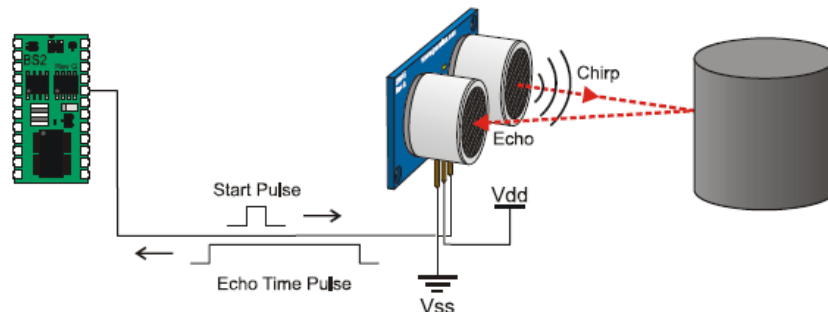
Web Site: [www.parallax.com](http://www.parallax.com)  
 Forums: [forums.parallax.com](http://forums.parallax.com)  
 Sales: [sales@parallax.com](mailto:sales@parallax.com)  
 Technical: [support@parallax.com](mailto:support@parallax.com)

Office: (916) 624-8333  
 Fax: (916) 624-8003  
 Sales: (888) 512-1024  
 Tech Support: (888) 997-8267

## PING)))™ Ultrasonic Distance Sensor (#28015)

The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, SX or Propeller chip, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



### Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2  $\mu$ s min, 5  $\mu$ s typ.
- Echo pulse: positive TTL pulse, 115  $\mu$ s minimum to 18.5 ms maximum.
- RoHS Compliant

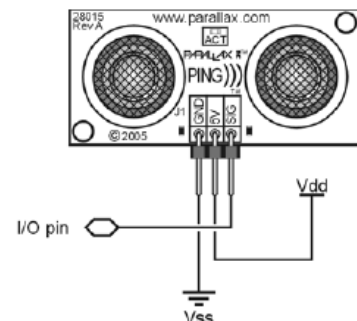
### Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

### Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #805-000012).







# SHARP

## GP2Y0A21YK

Optoelectronic Device

### FEATURES

- Digital Output
- LED Pulse Cycle Duration: 32 ms
- Range: 10 to 80 cm
- Typical response time: 39 ms
- Typical start up delay: 44 ms
- Average Current Consumption: 30 mA
- Detection Area Diameter @ 80 cm: 12 cm

### DESCRIPTION

The GP2Y0A21YK is a wide-angle distance measuring sensor.

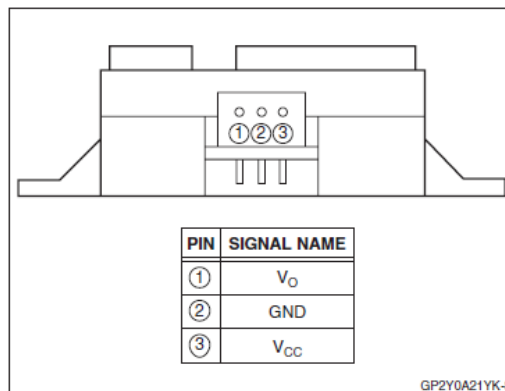


Figure 1. Pinout

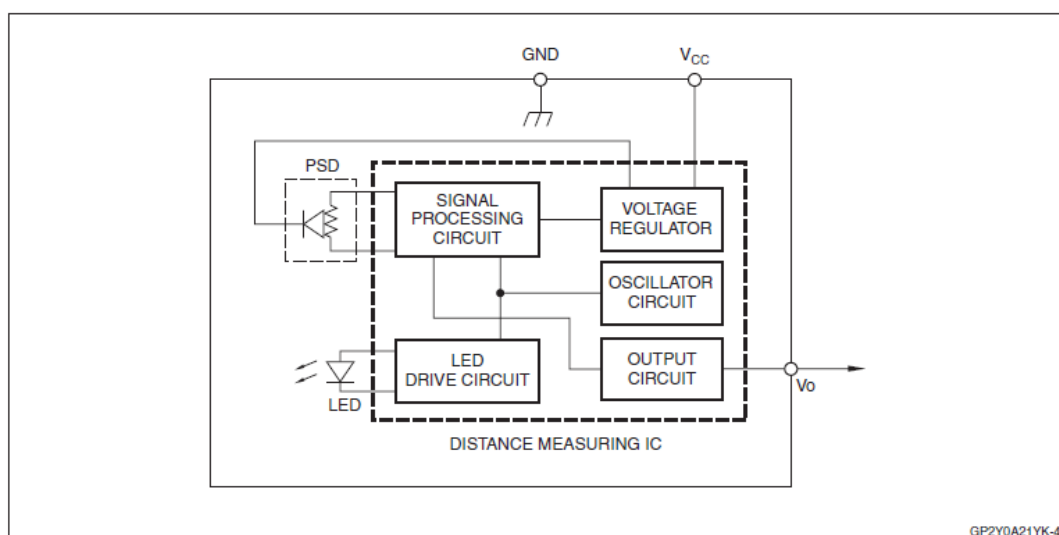


Figure 2. Block Diagram

## ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings

$T_a = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{ VDC}$

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	$V_{CC}$	-0.3 to +7	V
Output Terminal Voltage	$V_O$	-0.3 to ( $V_{CC} + 0.3$ )	V
Operating Temperature	$T_{opr}$	-10 to +60	$^\circ\text{C}$
Storage Temperature	$T_{stg}$	-40 to +70	$^\circ\text{C}$

### Operating Supply Voltage

PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	$V_{CC}$	4.5 to 5.5	V

### Electro-optical Characteristics

$T_a = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{ VDC}$

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTES
Measuring Distance Range	$\Delta L$		10	-	80	cm	1, 2
Output Terminal Voltage	$V_O$	$L = 80\text{ cm}$	0.25	0.4	0.55	V	1, 2
Output Voltage Difference	$\Delta V_O$	Output change at $\Delta L$ (80 cm - 10 cm)	1.65	1.9	2.15	V	1, 2
Average Supply Current	$I_{CC}$	$L = 80\text{ cm}$	-	30	40	mA	1, 2

#### NOTES:

- Using reflective object: white paper (made by Kodak Co. Ltd.) gray cards R-27, white face, reflective ratio: 90%
- $L$  = Distance to reflective object.

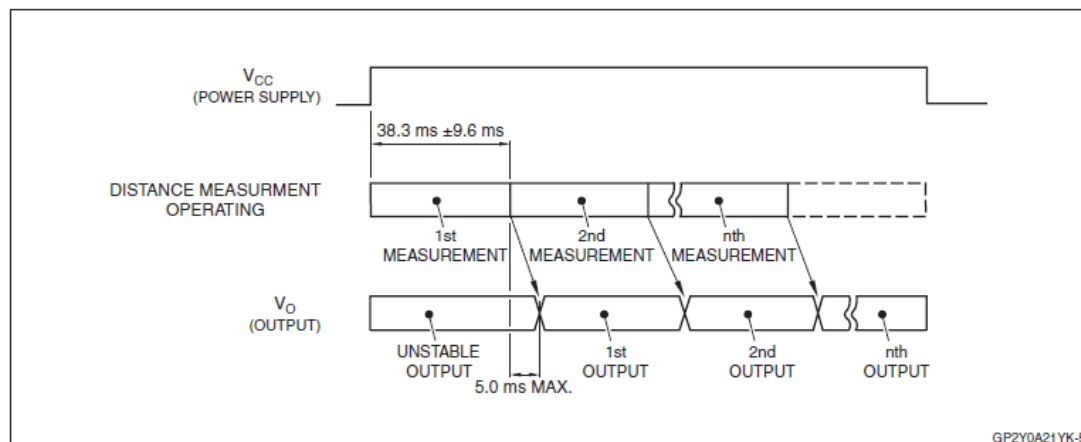


Figure 3. Timing Diagram

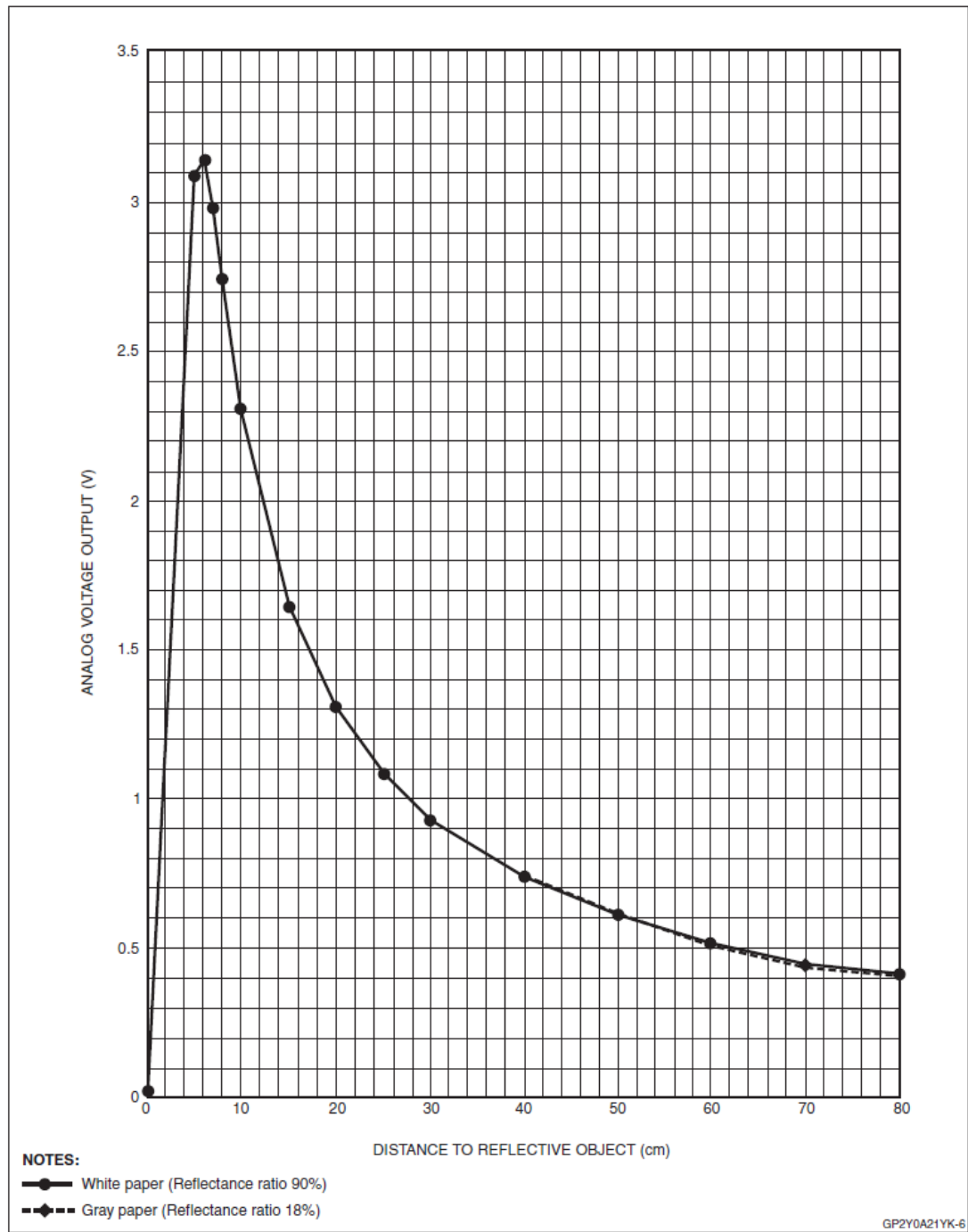
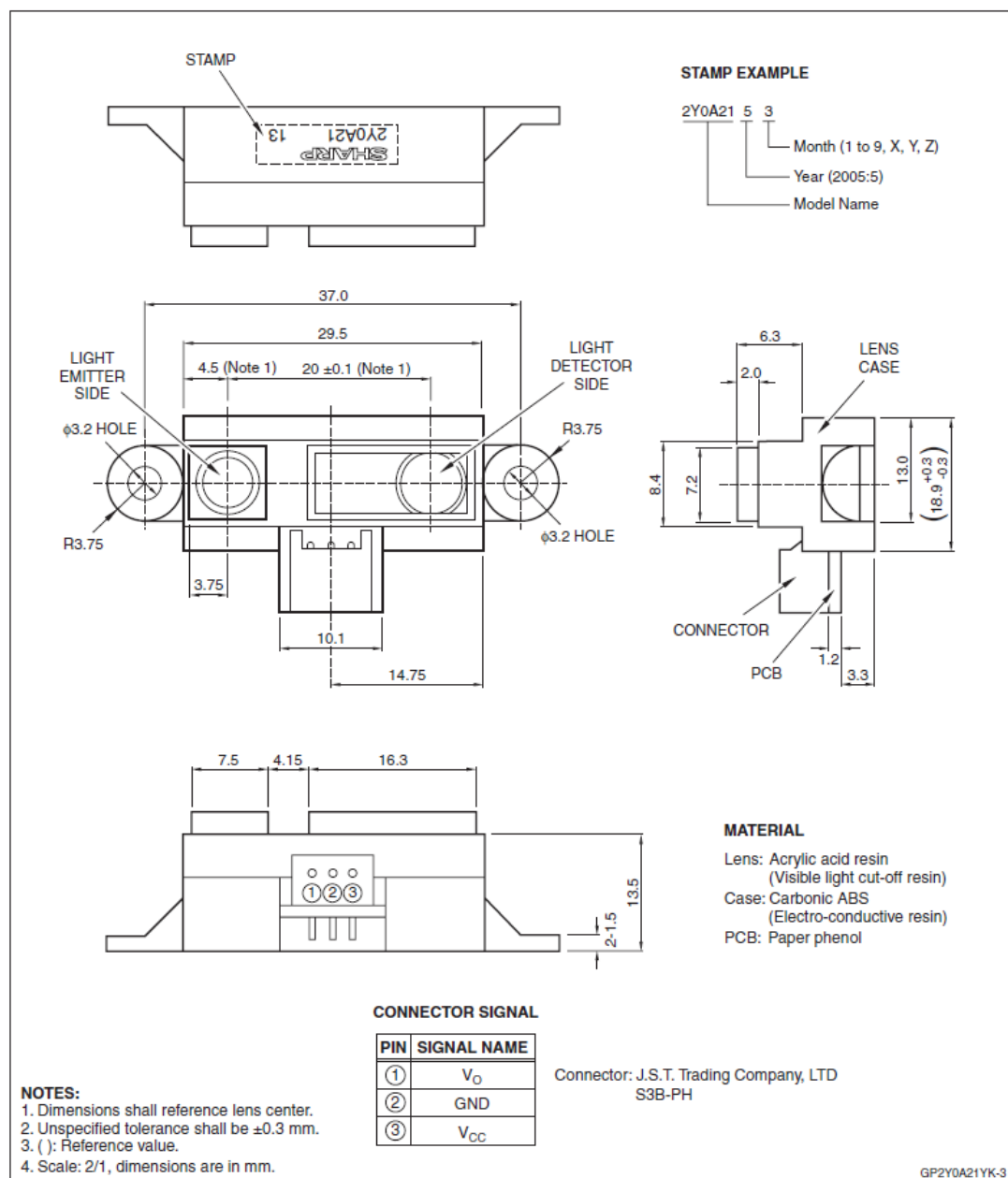


Figure 4. GP2Y0A021YK Example of Output Distance Characteristics

## PACKAGE SPECIFICATIONS





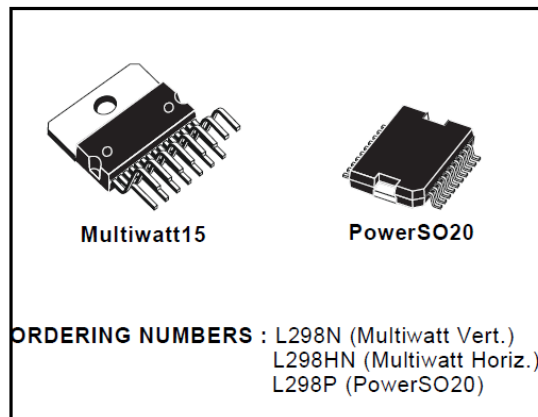
# L298

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

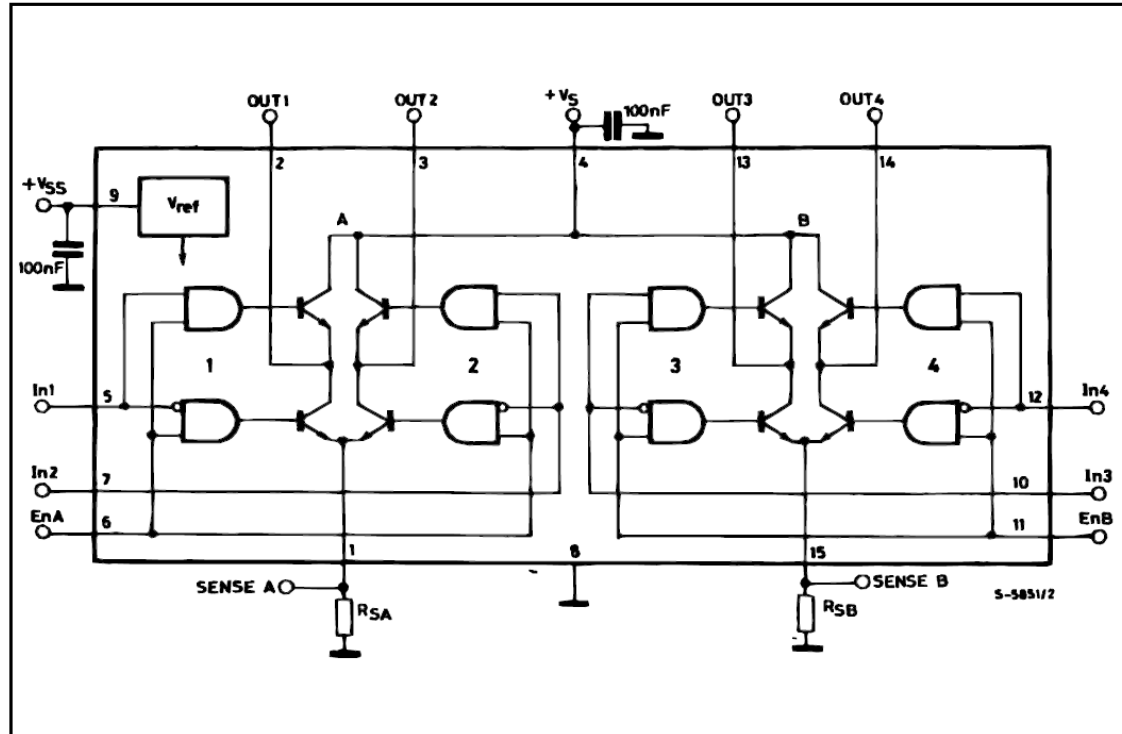
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



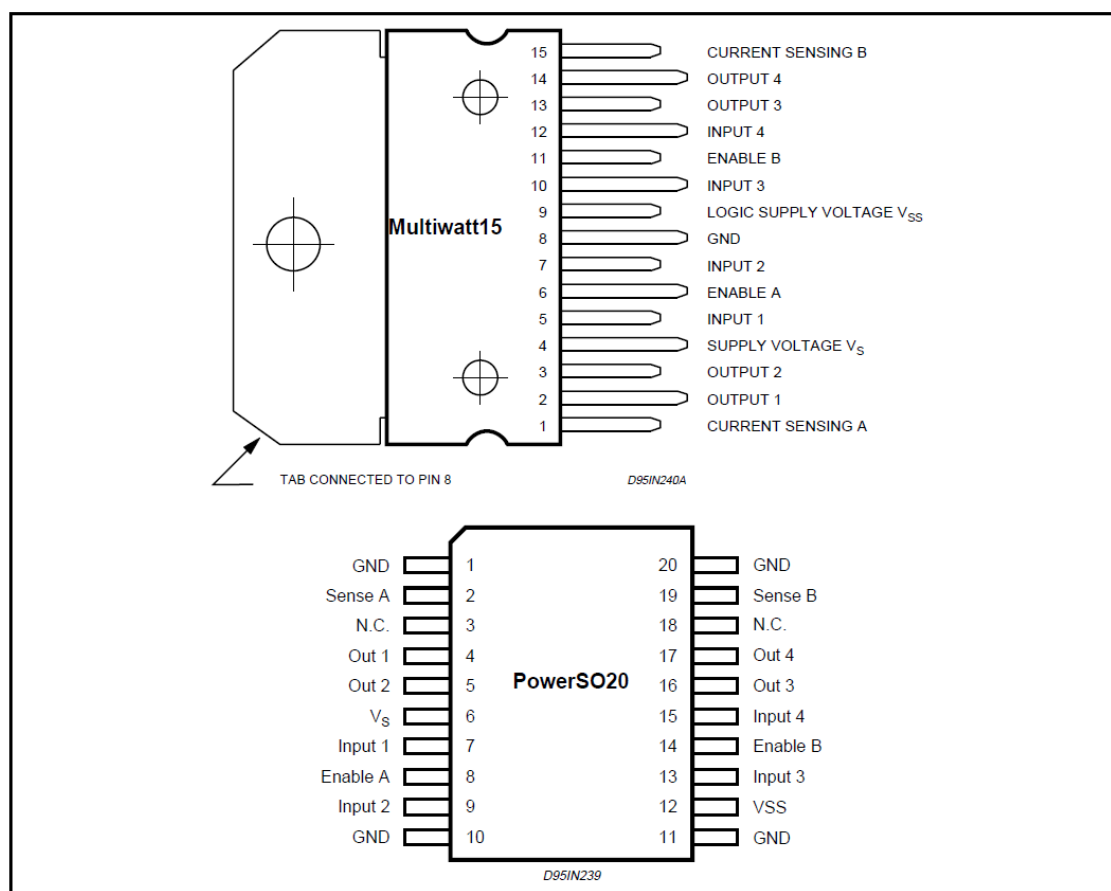
nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

### BLOCK DIAGRAM



**L298****ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_o$	Peak Output Current (each Channel)		
	– Non Repetitive ( $t = 100\mu s$ )	3	A
	– Repetitive (80% on –20% off; $t_{on} = 10ms$ )	2.5	A
	– DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)****THERMAL DATA**

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max. –	3	$^\circ C/W$
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max. 13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate



**PIN FUNCTIONS** (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	$V_S$	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	$V_{SS}$	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
—	3;18	N.C.	Not Connected

**ELECTRICAL CHARACTERISTICS** ( $V_S = 42V$ ;  $V_{SS} = 5V$ ,  $T_j = 25^\circ C$ ; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_S$	Supply Voltage (pin 4)	Operative Condition	$V_{IH} + 2.5$		46	V
$V_{SS}$	Logic Supply Voltage (pin 9)		4.5	5	7	V
$I_S$	Quiescent Supply Current (pin 4)	$V_{en} = H$ ; $I_L = 0$ $V_i = L$ $V_i = H$		13 50	22 70	mA mA
		$V_{en} = L$ $V_i = X$			4	mA
$I_{SS}$	Quiescent Current from $V_{SS}$ (pin 9)	$V_{en} = H$ ; $I_L = 0$ $V_i = L$ $V_i = H$		24 7	36 12	mA mA
		$V_{en} = L$ $V_i = X$			6	mA
$V_{IL}$	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
$V_{IH}$	Input High Voltage (pins 5, 7, 10, 12)		2.3		$V_{SS}$	V
$I_{IL}$	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_i = L$			-10	$\mu A$
$I_{IH}$	High Voltage Input Current (pins 5, 7, 10, 12)	$V_i = H \leq V_{SS} - 0.6V$		30	100	$\mu A$
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		$V_{SS}$	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	$\mu A$
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	$\mu A$
$V_{CEsat} (H)$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat} (L)$	Sink Saturation Voltage	$I_L = 1A$ (5) $I_L = 2A$ (5)	0.85	1.2 1.7	1.6 2.3	V V
$V_{CEsat}$	Total Drop	$I_L = 1A$ (5) $I_L = 2A$ (5)	1.80		3.2 4.9	V V
$V_{sens}$	Sensing Voltage (pins 1, 15)		-1 (1)		2	V



## LPC2141/42/44/46/48

Single-chip 16-bit/32-bit microcontrollers; up to 512 kB flash with ISP/IAP, USB 2.0 full-speed device, 10-bit ADC and DAC

Rev. 5 — 12 August 2011

Product data sheet

### 1. General description

The LPC2141/42/44/46/48 microcontrollers are based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine the microcontroller with embedded high-speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty.

Due to their tiny size and low power consumption, LPC2141/42/44/46/48 are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. Serial communications interfaces ranging from a USB 2.0 Full-speed device, multiple UARTs, SPI, SSP to I<sup>2</sup>C-bus and on-chip SRAM of 8 kB up to 40 kB, make these devices very well suited for communication gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWM channels and 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers suitable for industrial control and medical systems.

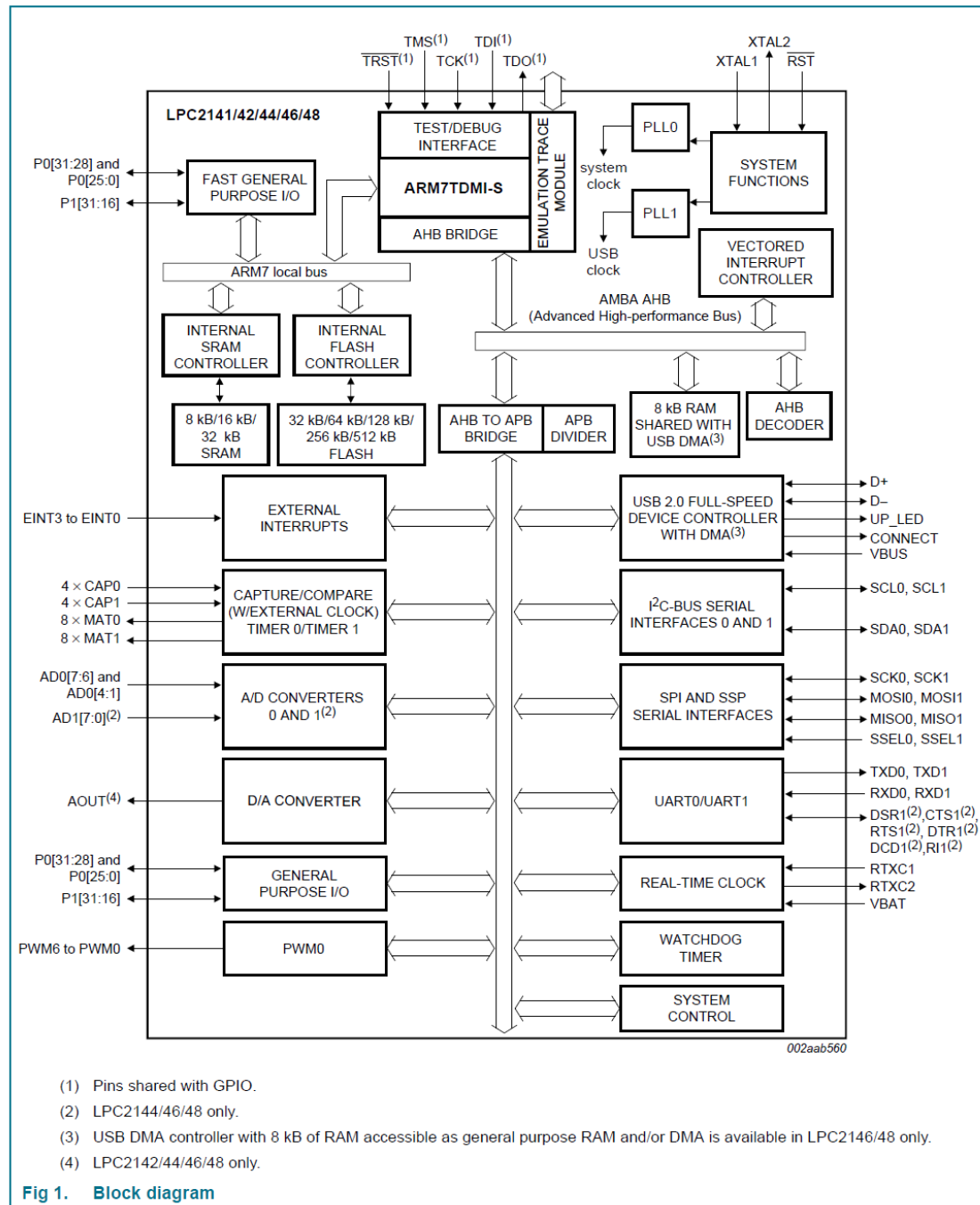
### 2. Features and benefits

#### 2.1 Key features

- 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory. 128-bit wide interface/accelerator enables high-speed 60 MHz operation.
- In-System Programming/In-Application Programming (ISP/IAP) via on-chip boot loader software. Single flash sector or full chip erase in 400 ms and programming of 256 B in 1 ms.
- EmbeddedICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip RealMonitor software and high-speed tracing of instruction execution.
- USB 2.0 Full-speed compliant device controller with 2 kB of endpoint RAM. In addition, the LPC2146/48 provides 8 kB of on-chip RAM accessible to USB by DMA.
- One or two (LPC2141/42 vs. LPC2144/46/48) 10-bit ADCs provide a total of 6/14 analog inputs, with conversion times as low as 2.44  $\mu$ s per channel.
- Single 10-bit DAC provides variable analog output (LPC2142/44/46/48 only).
- Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input.



## 4. Block diagram



### 6.3 On-chip static RAM

On-chip static RAM may be used for code and/or data storage. The SRAM may be accessed as 8-bit, 16-bit, and 32-bit. The LPC2141, LPC2142/44 and LPC2146/48 provide 8 kB, 16 kB and 32 kB of static RAM respectively.

In case of LPC2146/48 only, an 8 kB SRAM block intended to be utilized mainly by the USB can also be used as a general purpose RAM for data storage and code storage and execution.

### 6.4 Memory map

The LPC2141/42/44/46/48 memory map incorporates several distinct regions, as shown in [Figure 5](#).

In addition, the CPU interrupt vectors may be remapped to allow them to reside in either flash memory (the default) or on-chip static RAM. This is described in [Section 6.19](#) “System control”.

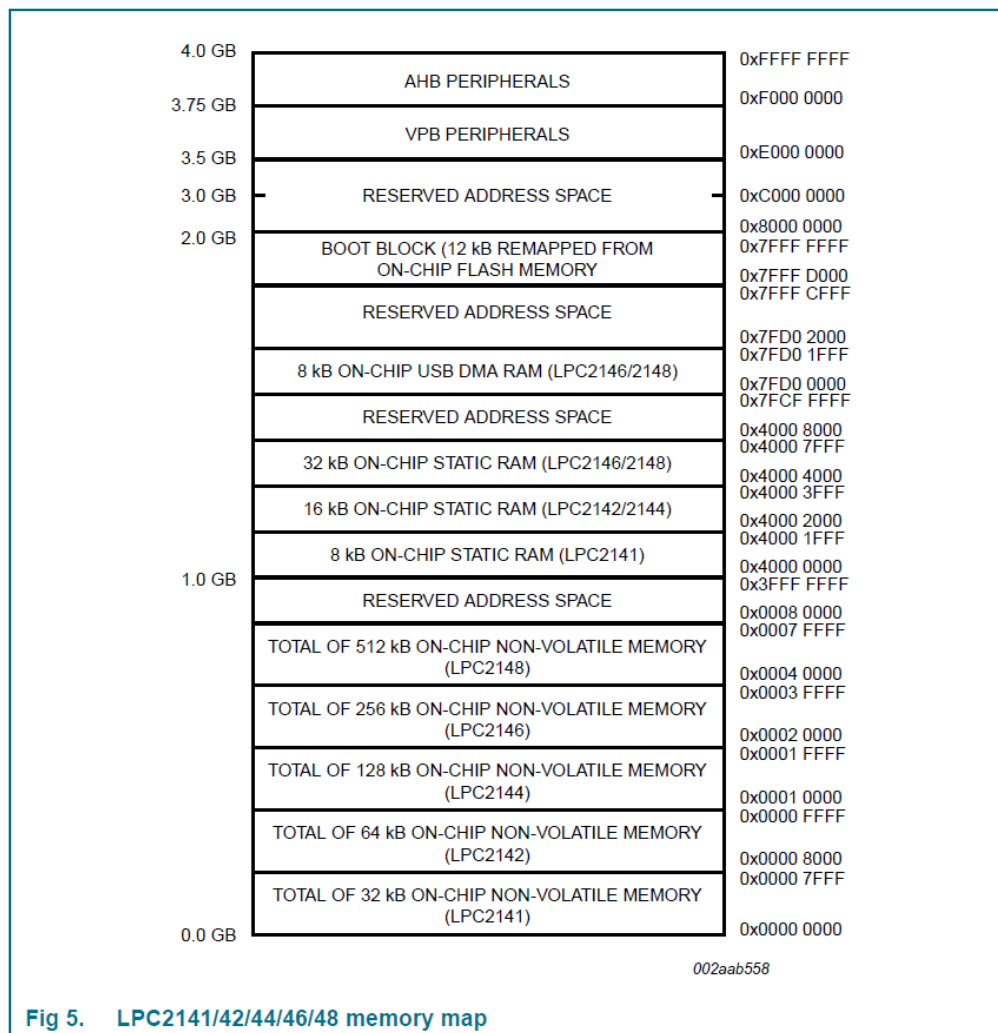


Fig 5. LPC2141/42/44/46/48 memory map

13. Package outline

LQFP64: plastic low profile quad flat package; 64 leads; body 10 x 10 x 1.4 mm

SOT314-2

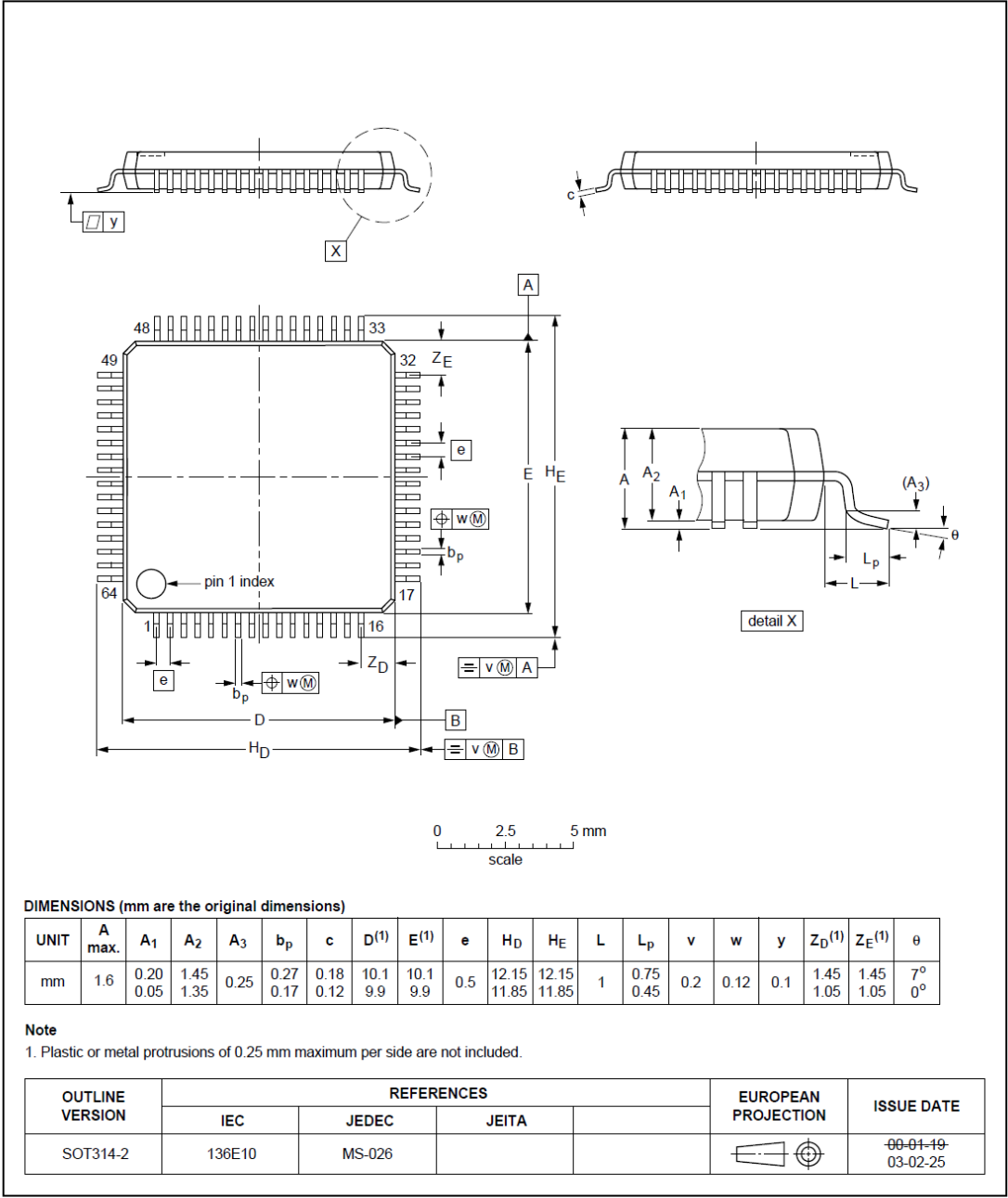


Fig 15. Package outline SOT314-2 (LQFP64)

## HS - 311 Standard

### Detailed Specifications

**Control System:** +Pulse Width Control 1500usec Neutral

**Required Pulse:** 3-5 Volt Peak to Peak Square Wave

**Operating Voltage:** 4.8-6.0 Volts

**Operating Temperature Range:** -20 to +60 Degree C

**Operating Speed (4.8V):** 0.19sec/60° at no load

**Operating Speed (6.0V):** 0.15sec/60° at no load

**Stall Torque (4.8V):** 42 oz/in (3.0 kg/cm)

**Stall Torque (6.0V):** 49 oz/in (4.5 kg/cm)

**Current Drain (4.8V):** 7.4mA/idle, 160mA no load operating

**Current Drain (6.0V):** 7.7mA/idle, 180mA no load operating

**Dead Band Width:** 5usec

**Operating Angle:** 40° one side pulse traveling 400usec

**Direction:** Clockwise/Pulse Traveling 1500 to 1900usec

**Motor Type:** Cored Metal Brush

**Potentiometer Drive:** 4 Slider/Direct Drive

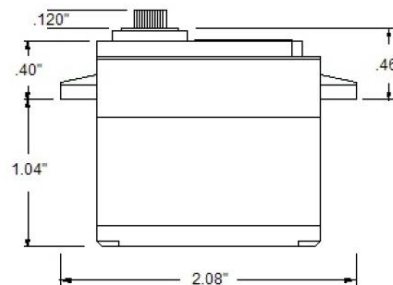
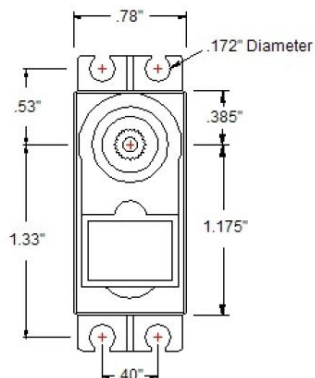
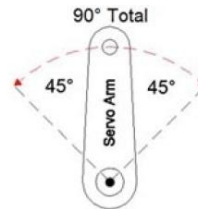
**Bearing Type:** Top/Resin Bushing

**Gear Type:** Nylon

**360 Modifiable:** [Yes](#)

**Connector Wire Length:** 11.81" (300mm)

**Weight:** 1.52oz (43g)







# PCA9306

Dual bidirectional I<sup>2</sup>C-bus and SMBus voltage-level translator

Rev. 6 — 25 November 2010

Product data sheet

## 1. General description

The PCA9306 is a dual bidirectional I<sup>2</sup>C-bus and SMBus voltage-level translator with an enable (EN) input, and is operational from 1.0 V to 3.6 V ( $V_{ref(1)}$ ) and 1.8 V to 5.5 V ( $V_{bias(ref)(2)}$ ).

The PCA9306 allows bidirectional voltage translations between 1.0 V and 5 V without the use of a direction pin. The low ON-state resistance ( $R_{on}$ ) of the switch allows connections to be made with minimal propagation delay. When EN is HIGH, the translator switch is on, and the SCL1 and SDA1 I/O are connected to the SCL2 and SDA2 I/O, respectively, allowing bidirectional data flow between ports. When EN is LOW, the translator switch is off, and a high-impedance state exists between ports.

The PCA9306 is not a bus buffer like the PCA9509 or PCA9517A that provide both level translation and physically isolates the capacitance to either side of the bus when both sides are connected. The PCA9306 only isolates both sides when the device is disabled and provides voltage level translation when active.

The PCA9306 can also be used to run two buses, one at 400 kHz operating frequency and the other at 100 kHz operating frequency. If the two buses are operating at different frequencies, the 100 kHz bus must be isolated when the 400 kHz operation of the other bus is required. If the master is running at 400 kHz, the maximum system operating frequency may be less than 400 kHz because of the delays added by the translator.

As with the standard I<sup>2</sup>C-bus system, pull-up resistors are required to provide the logic HIGH levels on the translator's bus. The PCA9306 has a standard open-collector configuration of the I<sup>2</sup>C-bus. The size of these pull-up resistors depends on the system, but each side of the translator must have a pull-up resistor. The device is designed to work with Standard-mode, Fast-mode and Fast-mode Plus I<sup>2</sup>C-bus devices in addition to SMBus devices. The maximum frequency is dependent on the RC time constant, but generally supports > 2 MHz.

When the SDA1 or SDA2 port is LOW, the clamp is in the ON-state and a low resistance connection exists between the SDA1 and SDA2 ports. Assuming the higher voltage is on the SDA2 port when the SDA2 port is HIGH, the voltage on the SDA1 port is limited to the voltage set by VREF1. When the SDA1 port is HIGH, the SDA2 port is pulled to the drain pull-up supply voltage ( $V_{pu(D)}$ ) by the pull-up resistors. This functionality allows a seamless translation between higher and lower voltages selected by the user without the need for directional control. The SCL1/SCL2 channel also functions as the SDA1/SDA2 channel.



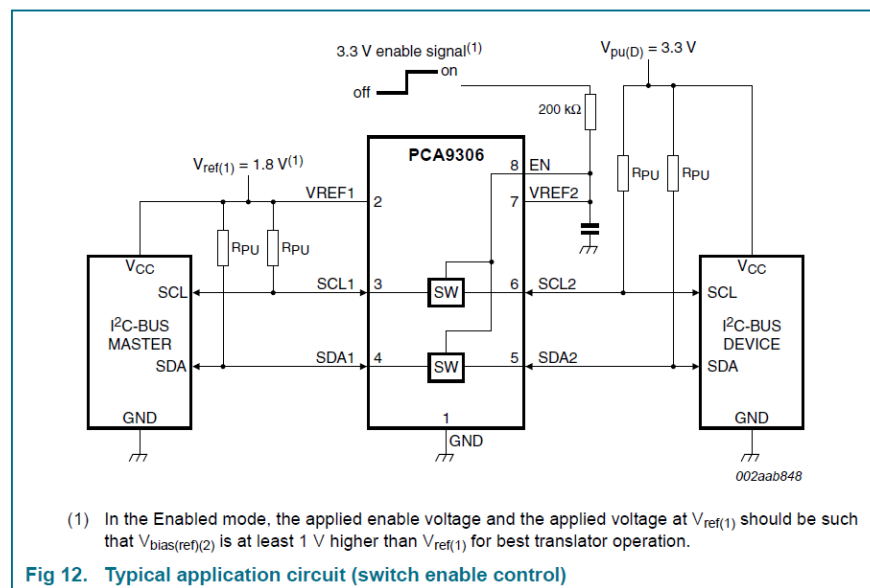
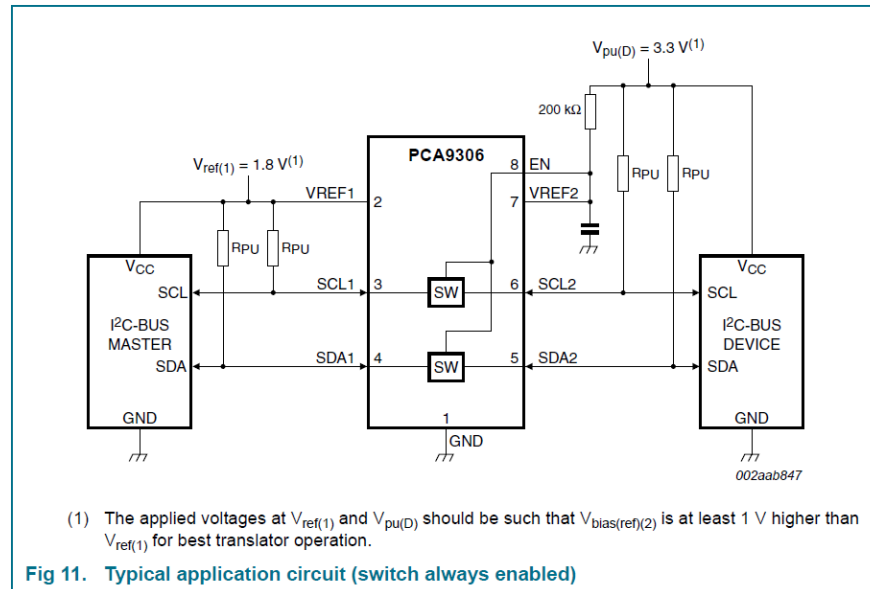


All channels have the same electrical characteristics and there is minimal deviation from one output to another in voltage or propagation delay. This is a benefit over discrete transistor voltage translation solutions, since the fabrication of the switch is symmetrical. The translator provides excellent ESD protection to lower voltage devices, and at the same time protects less ESD-resistant devices.

## 2. Features and benefits

- 2-bit bidirectional translator for SDA and SCL lines in mixed-mode I<sup>2</sup>C-bus applications
- Standard-mode, Fast-mode, and Fast-mode Plus I<sup>2</sup>C-bus and SMBus compatible
- Less than 1.5 ns maximum propagation delay to accommodate Standard-mode and Fast-mode I<sup>2</sup>C-bus devices and multiple masters
- Allows voltage level translation between:
  - ◆ 1.0 V  $V_{\text{ref}(1)}$  and 1.8 V, 2.5 V, 3.3 V or 5 V  $V_{\text{bias(ref)}(2)}$
  - ◆ 1.2 V  $V_{\text{ref}(1)}$  and 1.8 V, 2.5 V, 3.3 V or 5 V  $V_{\text{bias(ref)}(2)}$
  - ◆ 1.8 V  $V_{\text{ref}(1)}$  and 3.3 V or 5 V  $V_{\text{bias(ref)}(2)}$
  - ◆ 2.5 V  $V_{\text{ref}(1)}$  and 5 V  $V_{\text{bias(ref)}(2)}$
  - ◆ 3.3 V  $V_{\text{ref}(1)}$  and 5 V  $V_{\text{bias(ref)}(2)}$
- Provides bidirectional voltage translation with no direction pin
- Low 3.5  $\Omega$  ON-state connection between input and output ports provides less signal distortion
- Open-drain I<sup>2</sup>C-bus I/O ports (SCL1, SDA1, SCL2 and SDA2)
- 5 V tolerant I<sup>2</sup>C-bus I/O ports to support mixed-mode signal operation
- High-impedance SCL1, SDA1, SCL2 and SDA2 pins for EN = LOW
- Lock-up free operation
- Flow through pinout for ease of printed-circuit board trace routing
- ESD protection exceeds 2000 V HBM per JESD22-A114, 200 V MM per JESD22-A115, and 1000 V CDM per JESD22-C101
- Packages offered: SO8, TSSOP8, VSSOP8, XQFN8, XSON8, XSON8U

## 11. Application information



TSSOP8: plastic thin shrink small outline package; 8 leads; body width 3 mm; lead length 0.5 mm SOT505-2

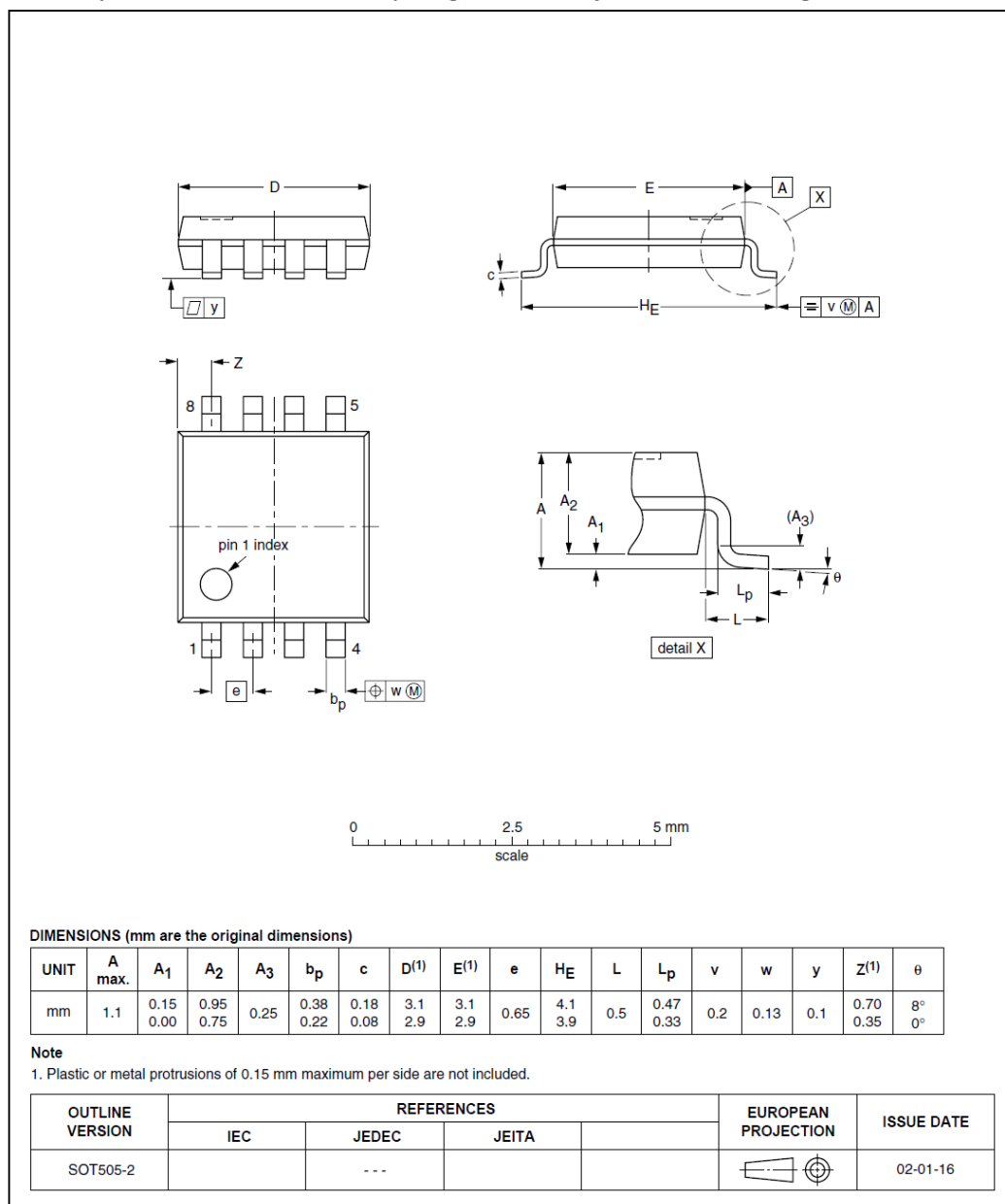
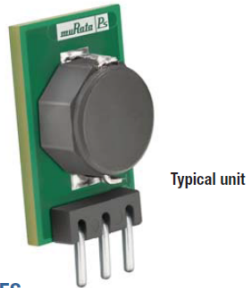


Fig 15. Package outline SOT505-2 (TSSOP8)



## Okami OKI-78SR Series

Fixed Output 1.5 Amp SIP DC/DC Converters



### FEATURES

- Ultra wide 7 to 36 VDC input range
- Fixed Outputs of 3.3 or 5 VDC up to 1.5 Amps
- Vertical or horizontal SIP-mount, small footprint package
- "No heat sink" direct replacement for 3-terminal 78xx-series linear regulators
- High efficiency with no external components
- Short circuit protection
- Outstanding thermal derating performance
- UL/EN/IEC 60950-1, 2nd Edition safety approvals (pending)

### PRODUCT OVERVIEW

Fabricated on a 0.41 by 0.65 inch (10.4 by 16.5 mm) Single Inline Package (SIP) module, the OKI-78SR series are non-isolated switching regulator (SR) DC/DC power converters for embedded applications. The fixed single output converters offer both tight regulation and high efficiency directly at the power usage site and are a direct plug-in replacement for TO-220 package 78xx series linear regulators. Typically, no extra outside components are required.

Two nominal output voltages are offered (3.3 and 5 VDC), each with 1.5 Amp maximum output.

Based on fixed-frequency buck switching topology, the high efficiency means very low heat and little electrical noise, requiring no external components. The ultra wide input range is 7 to 36 Volts DC.

Protection features include input undervoltage and short circuit protection, overcurrent and over temperature shut down. The OKI-78SR is designed to meet all standards approvals. RoHS-6 (no lead) hazardous material compliance is specified as standard.

Contents	Page
Description, Connection Diagram, Photograph	1
Ordering Guide, Model Numbering, Product Labeling	2
Detailed Electrical Specifications	3
Mechanical Specifications, Input/Output Pinout	4
Performance Data and Oscillograms	6
Soldering Guidelines, Technical Notes	10

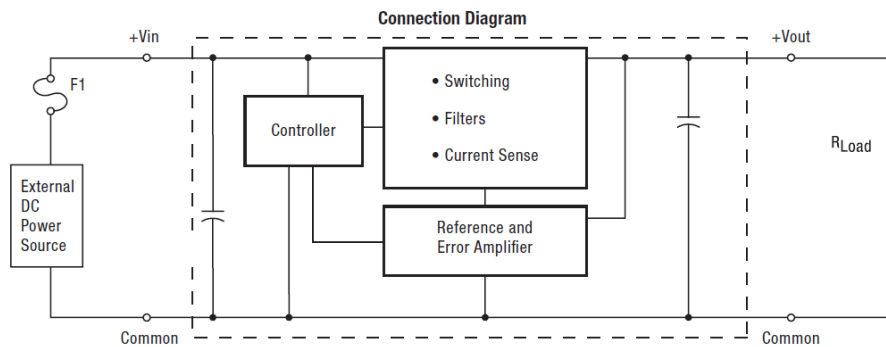


Figure 1. OKI-78SR

Note: Murata Power Solutions strongly recommends an external input fuse, F1. See specifications.

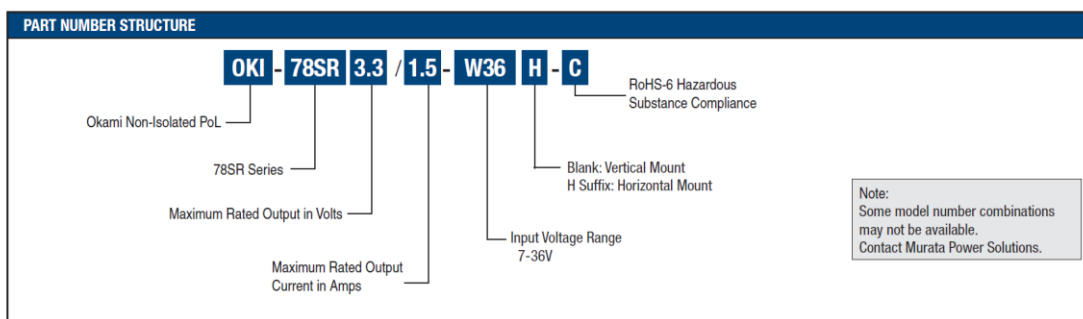


For full details go to  
[www.murata-ps.com/rohs](http://www.murata-ps.com/rohs)



FUNCTIONAL SPECIFICATIONS SUMMARY AND ORDERING GUIDE													
Root Model	Output						Input				Efficiency		Package ①
	V <sub>OUT</sub> (Volts)	I <sub>OUT</sub> (Amps max)	Power (Watts)	R/N (mVp-p)	Regulation (Max.)		V <sub>IN</sub> Nom. (Volts)	Range (Volts)	I <sub>IN</sub> , no load (mA)	I <sub>IN</sub> , full load (Amps)			
				Max.	Line	Load					Min.	Typ.	
OKI-78SR-3.3/1.5-W36-C	3.3	1.5	4.95	40	±0.25%	±0.25%	24	7-36	5	0.48	84%	85.5%	0.41 x 0.65 x 0.3 (10.4 x 16.5 x 7.62)
OKI-78SR-5/1.5-W36-C	5	1.5	7.5	75	±0.25%	±0.25%	24	7-36	5	0.69	89%	90.5%	0.41 x 0.65 x 0.3 (10.4 x 16.5 x 7.62)

① Dimensions are in inches (mm).

② All specifications are at nominal line voltage, V<sub>OUT</sub> = nominal and full load, +25 °C., with no external capacitor, unless otherwise noted.

### Product Label

Because of the small size of these products, the product label contains a character-reduced code to indicate the model number and manufacturing date code. Not all items on the label are always used. Please note that the label differs from the product photograph on page 1. Here is the layout of the label:

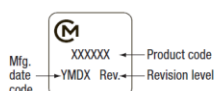


Figure 2. Label Artwork Layout

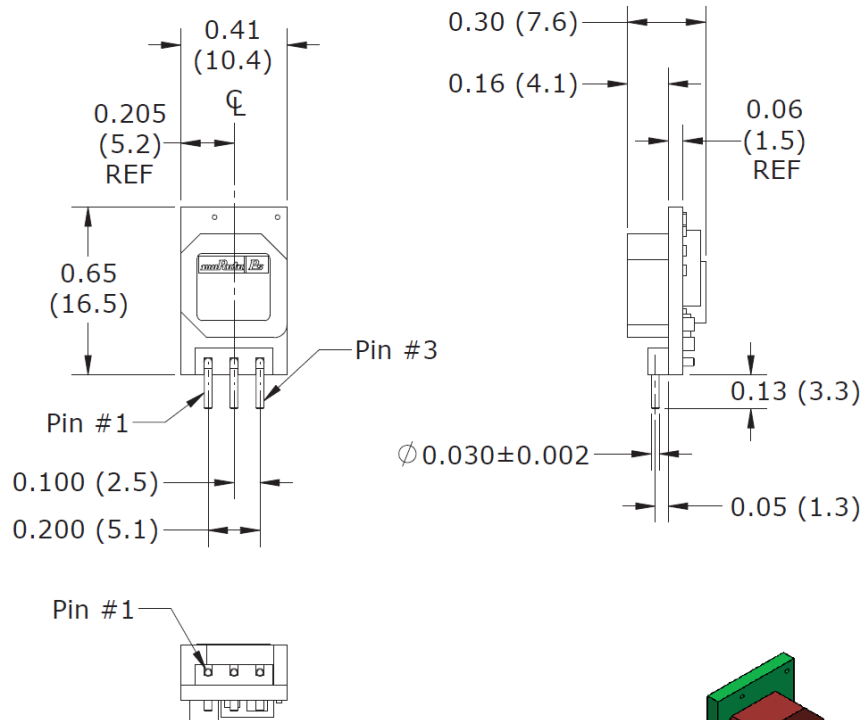
The label contains three rows of information:

First row – Murata Power Solutions logo  
 Second row – Model number product code (see table)  
 Third row – Manufacturing date code and revision level

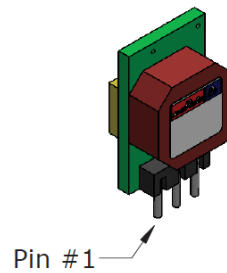
Model Number	Product Code
OKI-78SR-3.3/1.5-W36-C	I33115
OKI-78SR-5/1.5-W36-C	I50115
OKI-78SR-3.3/1.5-W36H-C	I33115H
OKI-78SR-5/1.5-W36H-C	I50115H

The manufacturing date code is four characters:

First character – Last digit of manufacturing year, example 2009  
 Second character – Month code (1 through 9 and 0 through D)  
 Third character – Day code (1 through 9 = 1 to 9, 10 = 0 and 11 through 31 = A through Z)  
 Fourth character – Manufacturing information

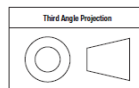
**MECHANICAL SPECIFICATIONS, VERTICAL MOUNT**

PIN MATERIAL: COPPER ALLOY  
 PIN FINISH: PURE MATTE TIN 100-300 u"  
 OVER 75-150 u" NICKEL



INPUT/OUTPUT CONNECTIONS OKI-78SR	
Pin	Function
1	Positive Input
2	Common (Ground)
3	Positive Output

Dimensions are in inches (mm shown for ref. only).



Tolerances (unless otherwise specified):  
 XX ± 0.02 (0.5)  
 XXX ± 0.010 (0.25)  
 Angles ± 1°

Components are shown for reference only.

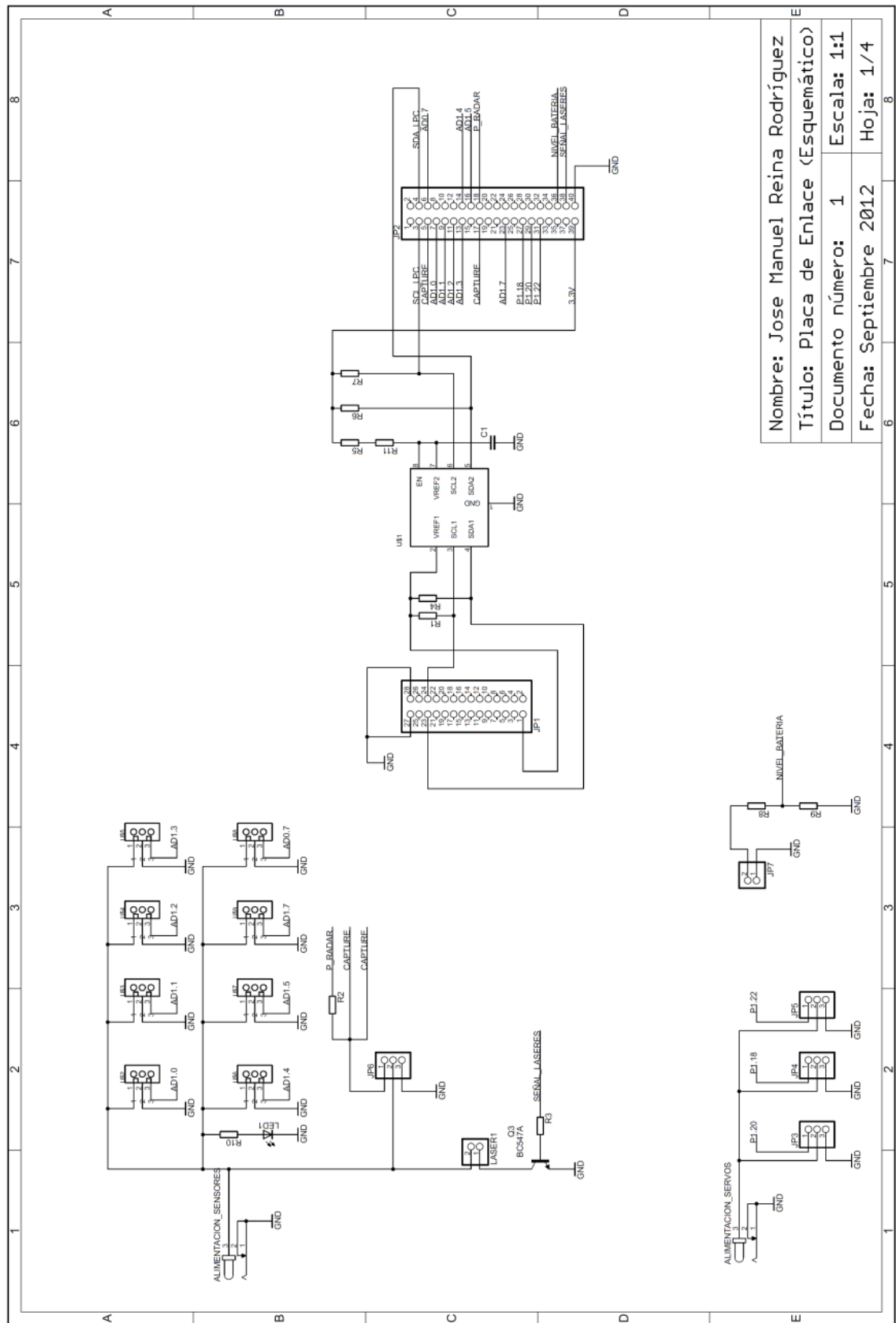
## **2.1. Código**

El código es de gran extensión, habiendo hojas de más de mil líneas por lo que se ha decidido que se adjunte en formato electrónico junto al libro

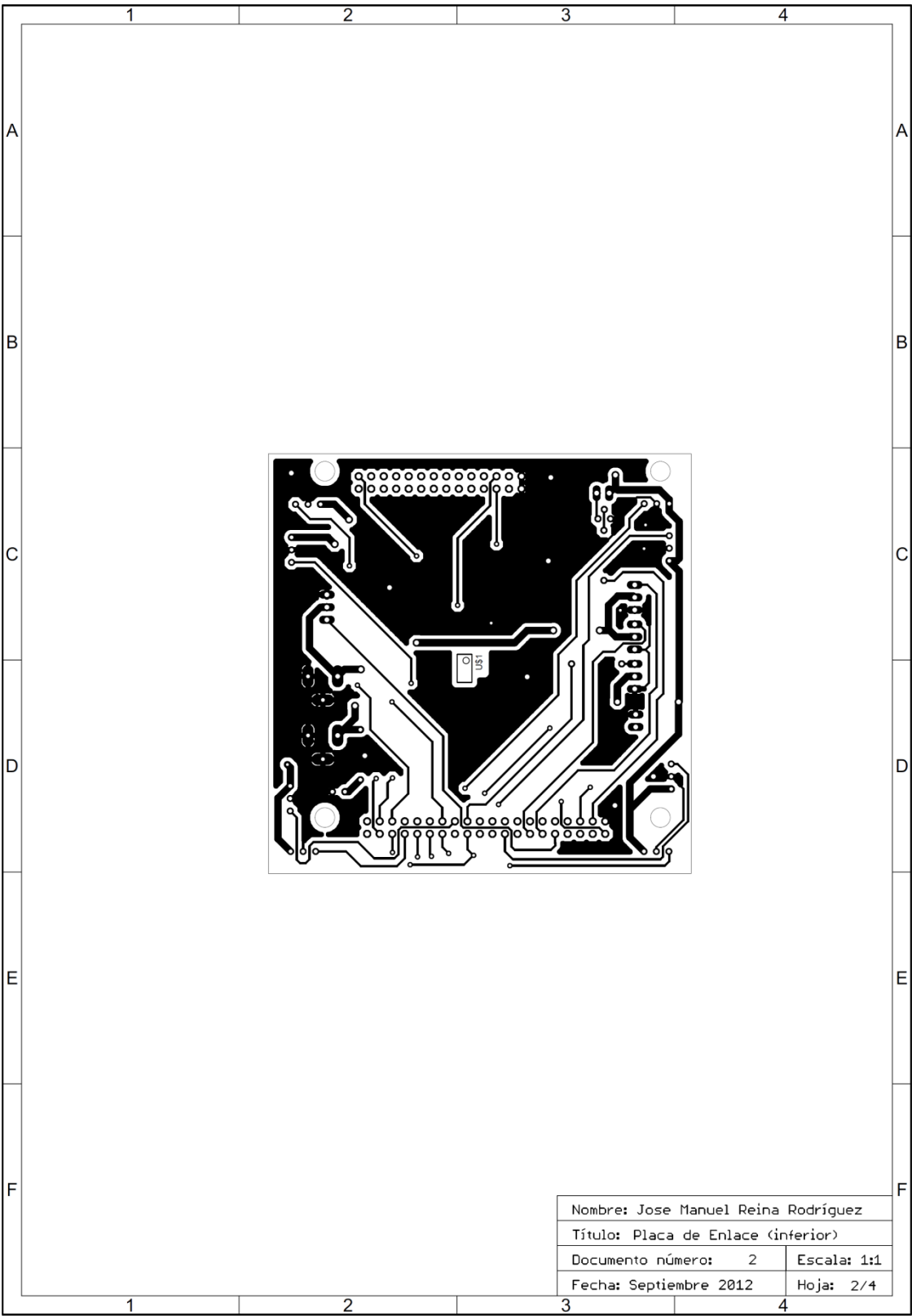




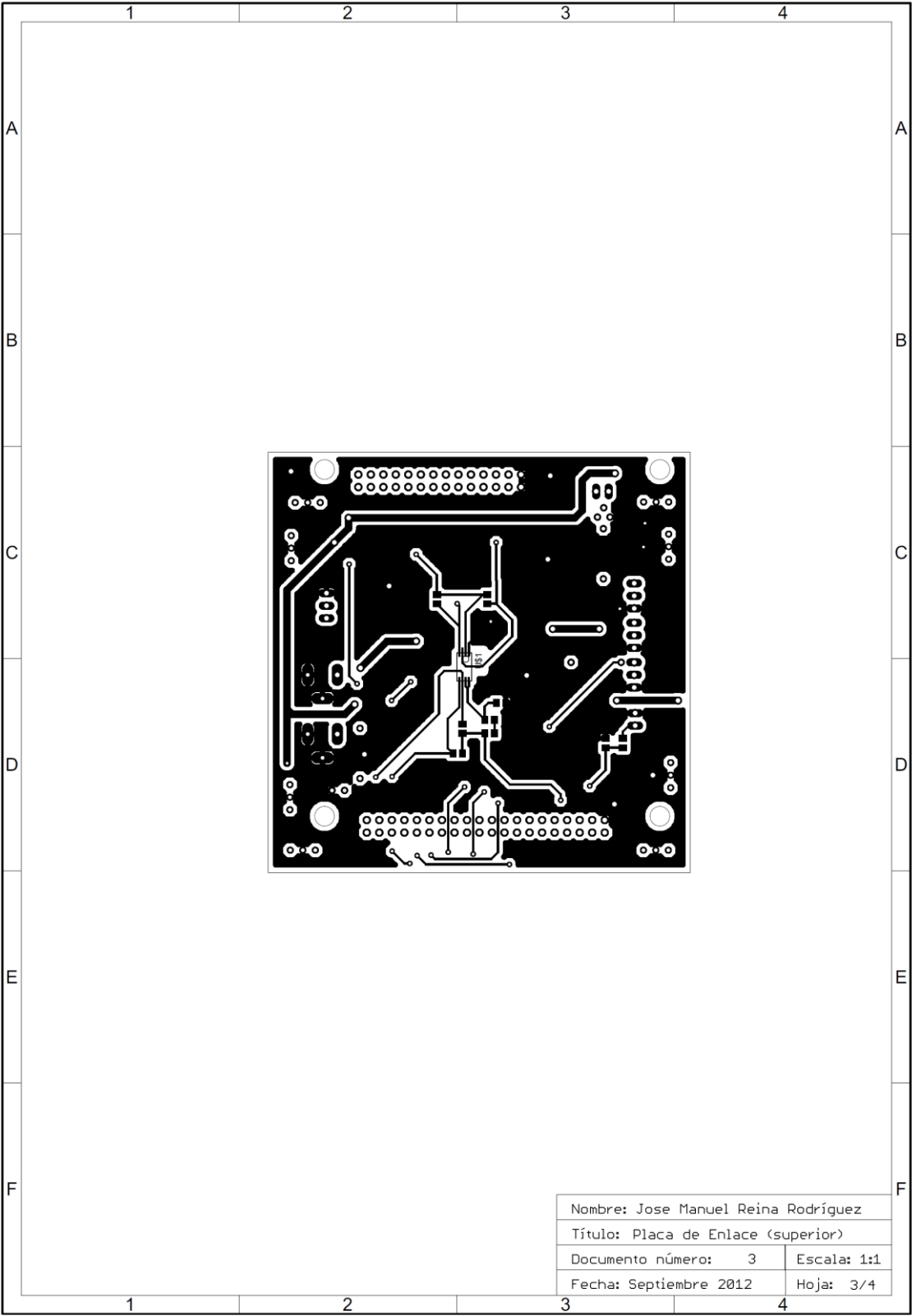
## 3. Planos



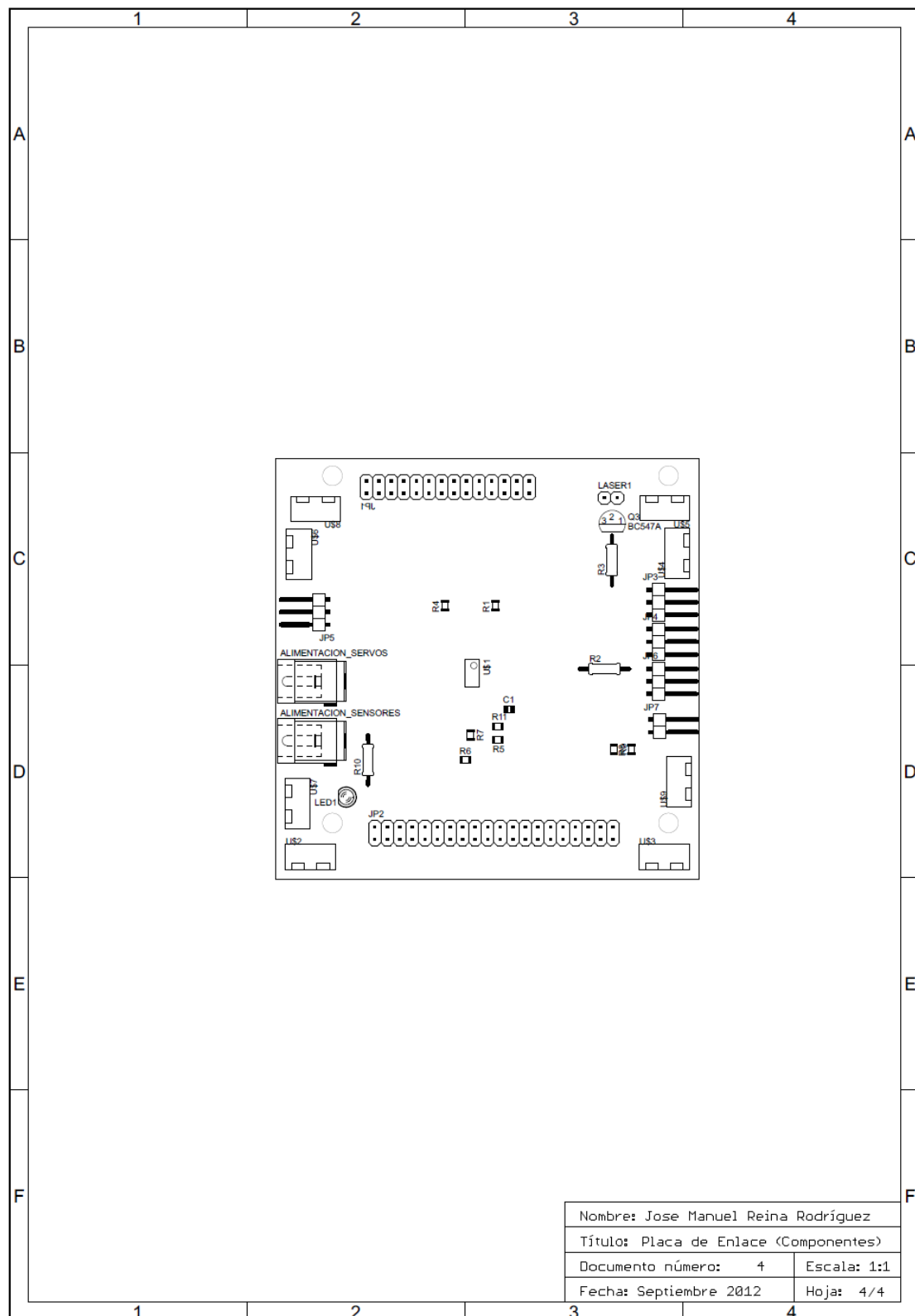






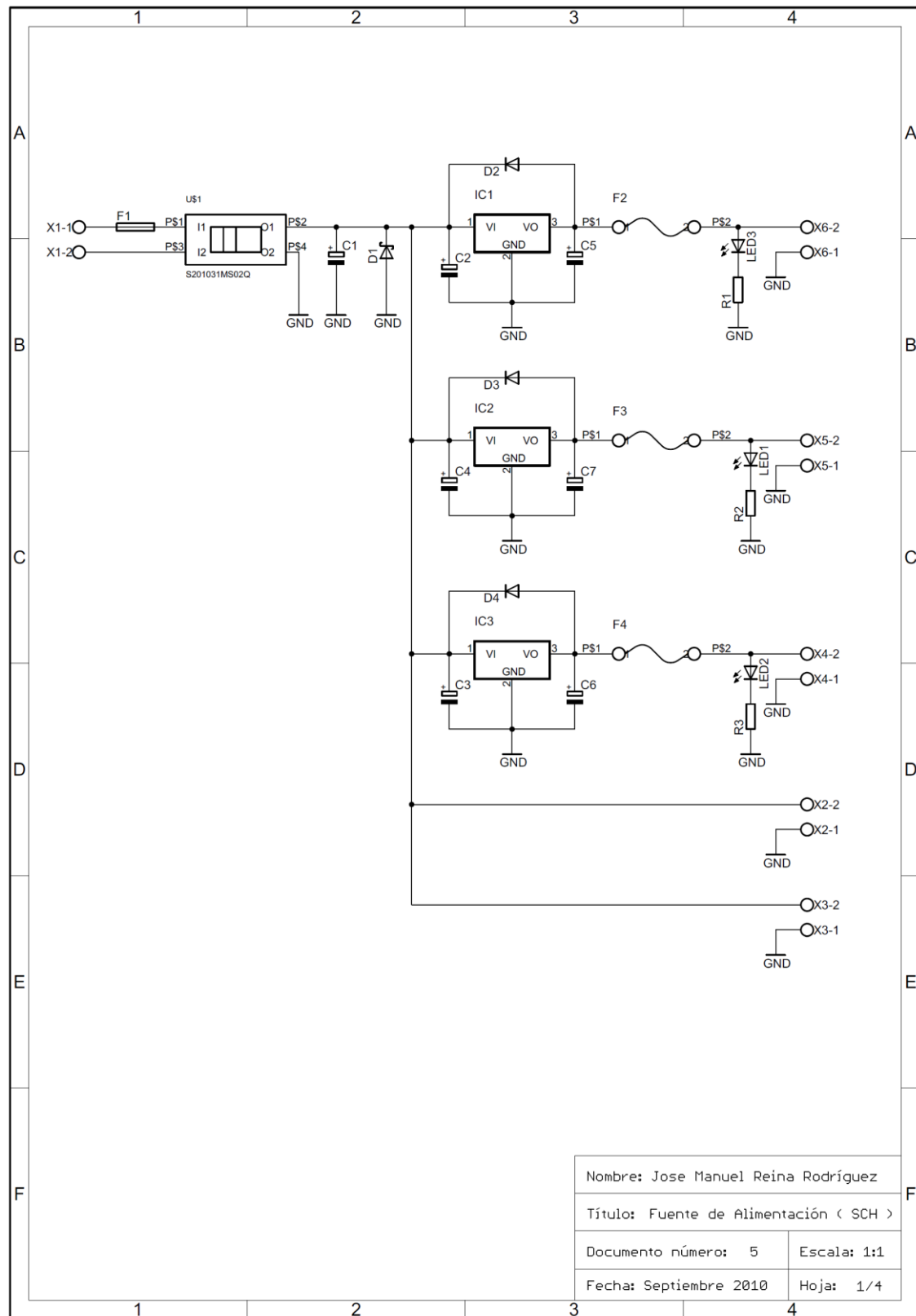




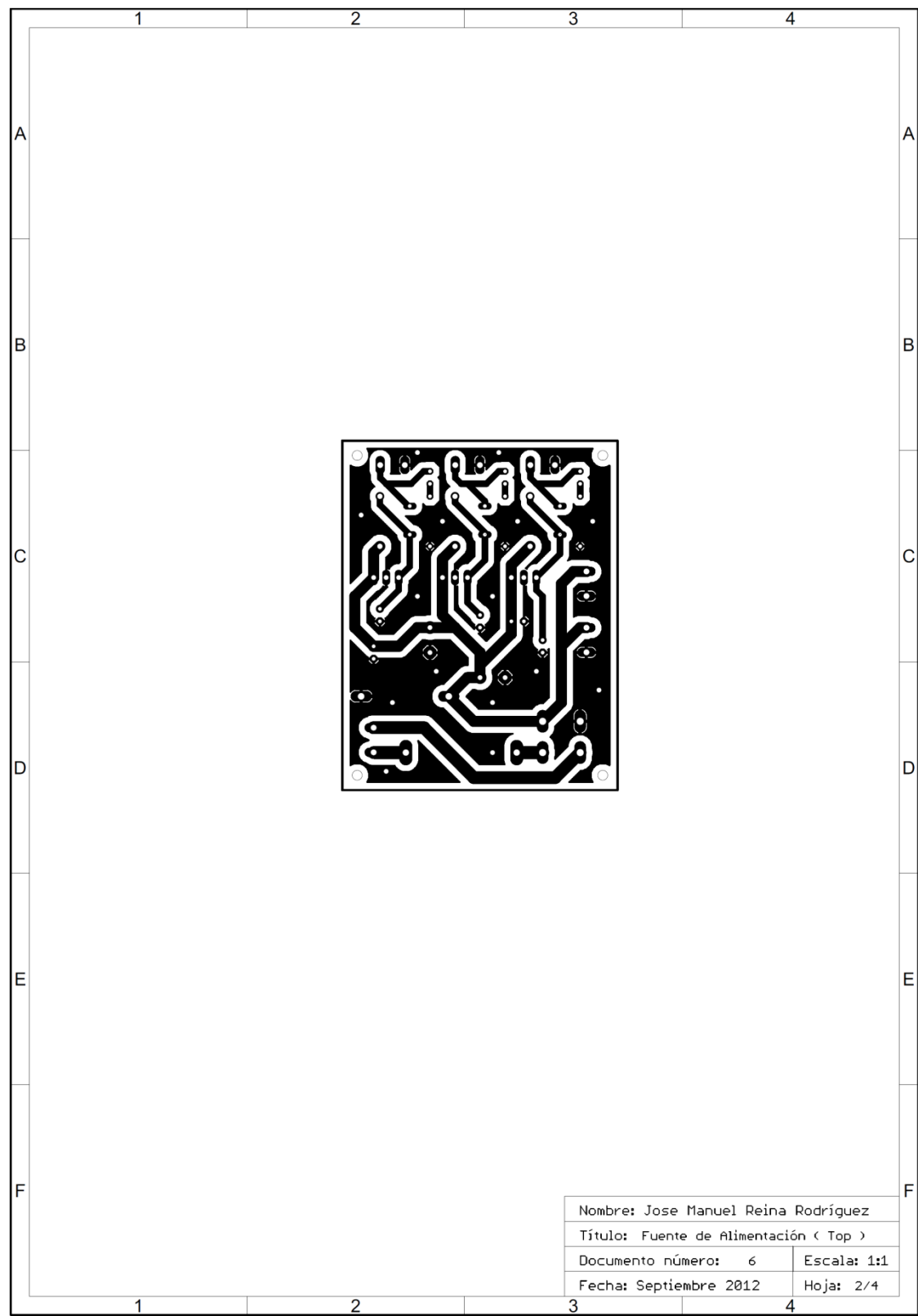




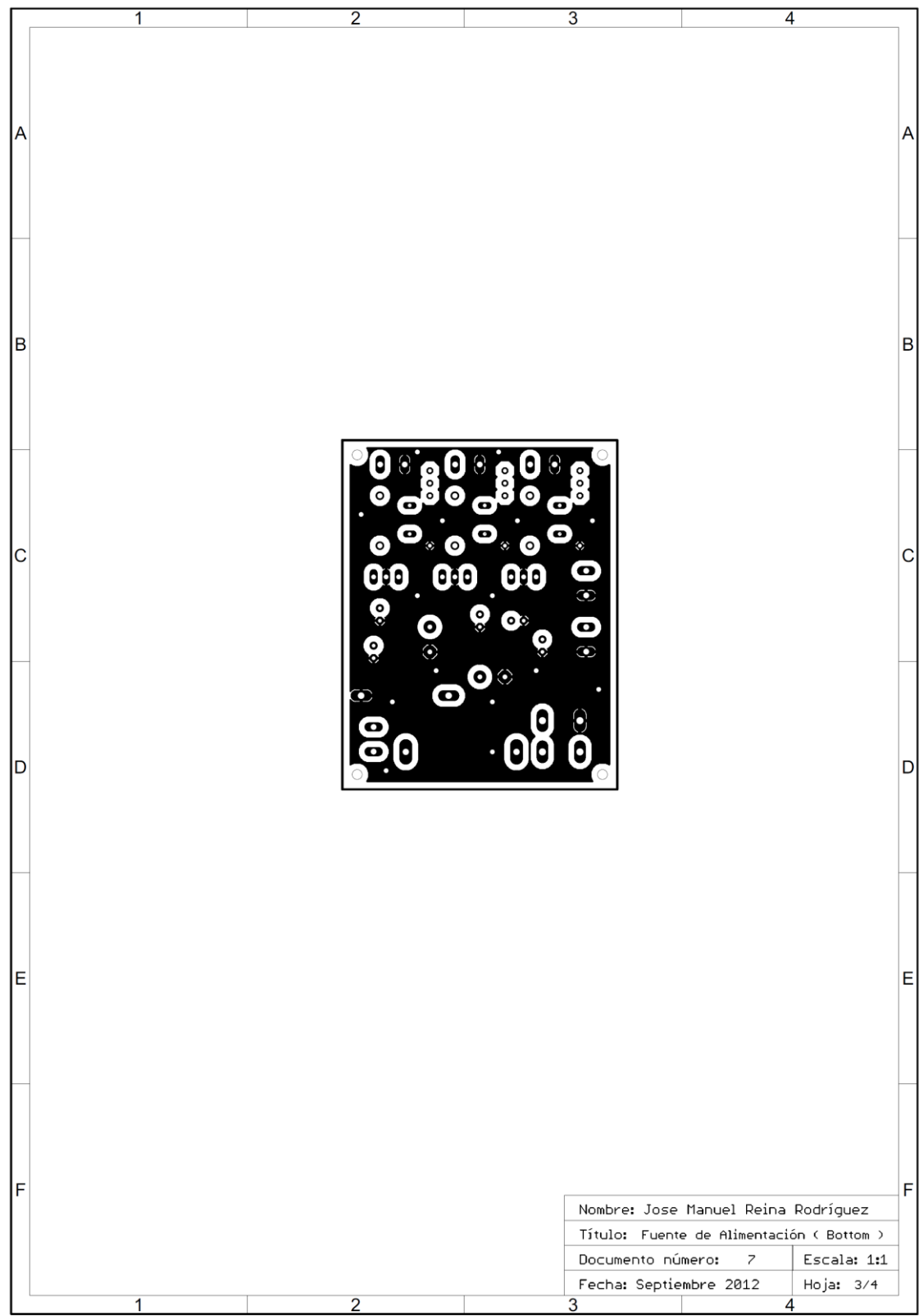




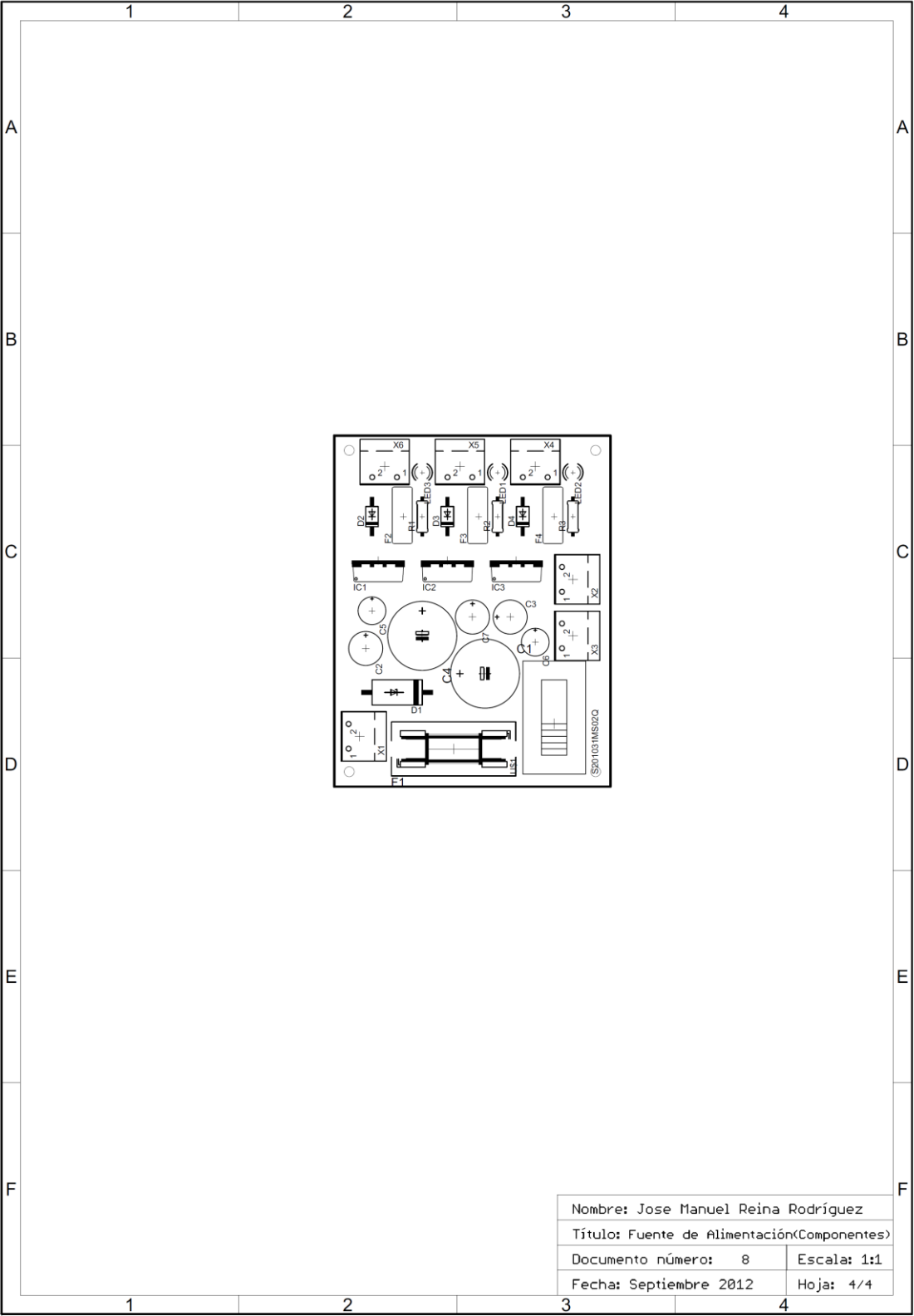












Nombre: Jose Manuel Reina Rodríguez	
Título: Fuente de Alimentación(Componentes)	
Documento número: 8	Escala: 1:1
Fecha: Septiembre 2012	Hoja: 4/4





## 4. Presupuesto

## 4.1. Cuadro de precios

### 4.1.1. Módulo L298

ID	Descripción	Unidades	Coste unitario	TOTAL
1	Módulo L298	1	6,58€/u	<b>6,58€</b>

### 4.1.2. Placa de ASA

ID	Descripción	Unidades	Coste unitario	TOTAL
2	Placa ASA	1	20€/u	<b>20€</b>

### 4.1.3. Beagleboard-xM

ID	Descripción	Unidades	Coste unitario	TOTAL
3	BeagleBoard, modelo xM	1	133€/u	<b>133€</b>

### 4.1.4. Placa de enlace

ID	Descripción	Unidades	Coste unitario	TOTAL
4	Tiras de pines poste largo, x40	1	0,30€/u	0,30€
5	Conector JKT de 3 pines, 2,54 mm	8	0,12€/u	0,96€
6	Resistencias SMD	6	0,01€/u	0,06€
7	Resistencias de agujero pasante	3	0,10€/u	0,30€
8	LED azul, 3 mm	1	0,15€/u	0,15€
9	Tira de pines acodado x40, 2,54 mm	1	0,30€/u	0,30€
10	Circuito integrado PCA9306	1	0,65€/u	0,65€
11	Conector JACK	2	0,20€/u	0,40€
12	Transistor BC547	1	0,10€/u	0,10€
13	Conector 2x20, 2,54 mm aislado	1	0,22€/u	0,22€
<b>SUMA TOTAL</b>				<b>3,44€</b>

## 4.1.5. Alimentación

### 4.1.5.1. Placa de alimentación

ID	Descripción	Unidades	Coste unitario	TOTAL
14	Convertor DC/DC OKI-78SR-5	3	3,52€/u	10,56€
15	Condensador 100 uf, 25V	2	0,21€/u	0,42€
16	Condensador 10 uf, 16V	2	0,67€/u	1,34€
17	Condensador 47 uf, 16V	1	2,22€/u	2,22€
18	Condensador 470 uf, 25V	1	0,79€/u	0,79€
19	Condensador 470 uf, 63V	1	0,83€/u	0,83€
20	Fusible 5A	1	0,268€/u	0,268€
21	Soporte fusible	1	0,57€/u	0,57€
22	Tapa de soporte de fusible	1	0,25€/u	0,25€
23	Diodo Schottky, 15V, 9A	1	2,80€/u	2,80€
24	Fusible Rearmable 1A, PTC	3	0,64€/u	1,92€
25	Conector Clema, 2 entradas	6	0,41€/u	2,46€
26	Interruptor, 6A	1	2,64€/u	2,64€
<b>SUMA TOTAL</b>				<b>27,06€</b>

### 4.1.5.2. Cables de alimentación

ID	Descripción	Unidades	Coste unitario	TOTAL
27	Conector JACK Hembra	4	1,13€/uu	4,52€
28	Cable 24 AWG, 1m	1	0,445€/metro	0,445€
<b>SUMA TOTAL</b>				<b>4,96€</b>

### 4.1.5.3. Conjunto batería y cargador

ID	Descripción	Unidades	Coste unitario	TOTAL
37	Batería LiPo, 5.000 mAh	1	19,91€/u	19,91€
38	Cargador de batería Imax B6	1	29,79€/u	29,79€
39	Tester batería LiPo	1	2,85€/u	2,85€
40	Conectores HXT	2	0,27€/u	0,54€
41	Cable azul, 14 AWG, 0,25cm	1	0,96€/metro	0,24€/0,25cm
42	Cable rojo, 18 AWG, 0,5m	1	0,89€/metro	0,445€/0,5m
<b>SUMA TOTAL</b>				<b>53,77€</b>

### 4.1.6. Servomotores

ID	Descripción	Unidades	Coste unitario	TOTAL
29	Servomotor Hitec Hs-311	3	12,50€/u	37,50€
30	Juego PAN and Titt	1	10€/u	10€
<b>SUMA TOTAL</b>				<b>47,50€</b>

### 4.1.7. Sensores

ID	Descripción	Unidades	Coste unitario	TOTAL
31	Parallax PING	1	36,50€/u	36,50€
32	Sharp GP240A214K	4	14,50€/u	58€
33	Sharp GP2D12	4	11,20€/u	44,80€
34	Láser	2	4,80€/u	9,60€
35	Conector JST	8	0,38€/u	3,04€
<b>SUMA TOTAL</b>				<b>151,94€</b>

### 4.1.8. Cámara

ID	Descripción	Unidades	Coste unitario	TOTAL
36	Cámara Logitech C210	1	19,90€/u	<b>19,90€</b>

### 4.1.9. Comunicación Wi-Fi

ID	Descripción	Unidades	Coste unitario	TOTAL
43	Router	1	18€/u	18€
44	Conector Wi-Fi USB	1	20€/u	20€
45	Alargador USB con soporte	1	3€/u	3€
<b>SUMA TOTAL</b>				<b>41€</b>

## 4.2. Presupuesto

- Presupuesto del Módulo L298 .....6,58 €
- Presupuesto de la placa de ASA.....20 €
- Presupuesto de la BeagleBoard-xM.....133 €
- Presupuesto de la placa enlace..... 3,44 €
- Presupuesto de alimentación global ..... 85,81 €
- Presupuesto de servomotores..... 47,50 €
- Presupuesto de sensores..... 151,94 €
- Presupuesto de cámara ..... 19,90 €
- Presupuesto de la comunicación Wi-Fi.....41 €
- Presupuesto horas de trabajo 1000horas x 20 €/hora..... 20000€

---

**Total presupuestario .....20508,17 €**

---

**I.V.A. (21%) .....4306,71 €**

---

**Total presupuestario .....24814,89 €**

---

